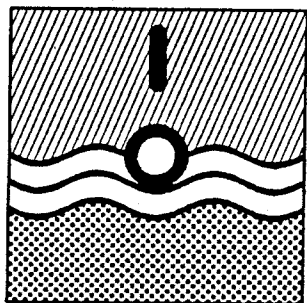


ΥΔΡΟΣΚΟΠΙΟ

ΠΡΟΓΡΑΜΜΑ STRIDE ΕΛΛΑΣ

ΔΗΜΙΟΥΡΓΙΑ ΕΘΝΙΚΗΣ ΤΡΑΠΕΖΑΣ
ΥΔΡΟΛΟΓΙΚΗΣ ΚΑΙ ΜΕΤΕΩΡΟΛΟΓΙΚΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ



HYDROSCOPE

STRIDE HELLAS PROGRAMME

DEVELOPMENT OF A NATIONAL DATA
BANK FOR HYDROLOGICAL AND
METEOROLOGICAL INFORMATION

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΟΜΕΑΣ ΥΔΑΤΙΚΩΝ ΠΟΡΩΝ,
ΥΔΡΑΥΛΙΚΩΝ ΚΑΙ ΘΑΛΑΣΣΙΩΝ ΕΡΓΩΝ

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
DEPARTMENT OF WATER RESOURCES,
HYDRAULIC AND MARITIME ENGINEERING

ΣΧΕΔΙΑΣΜΟΣ ΣΧΗΜΑΤΟΣ
ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

DATABASE SCHEMA DESIGN

*Ν. Παπακώστας
Ομάδα Πληροφορικής Ε.Μ.Π.*

Αριθμός τεύχους 1/15
Report number

ΑΘΗΝΑ - ΔΕΚΕΜΒΡΙΟΣ 1993
ATHENS - DECEMBER 1993

ΠΕΡΙΛΗΨΗ

Η παρούσα μελέτη αφορά στο σχεδιασμό και την υλοποίηση της βάσης δεδομένων του ΥΔΡΟΣΚΟΠΙΟΥ. Περιλαμβάνει καταγραφή, ομαδοποίηση και κατάταξη των διαφόρων αντικειμένων της ΒΔ, περιγραφή των συσχετίσεων μεταξύ τους, ανάλυση της λειτουργίας σε κατανεμημένο περιβάλλον, των προβλημάτων που αυτή συνεπάγεται και των λύσεων που δόθηκαν. Επίσης περιλαμβάνει λεπτομερή παρουσίαση του τρόπου με τον οποίο τα δεδομένα ως φυσικές οντότητες απεικονίζονται στα διάφορα αντικείμενα της ΒΔ και του τρόπου με τον οποίο αντιμετωπίζονται τα προβλήματα της χρέωσης, της ασφάλειας και της προσαρμογής των εργαλείων που χρησιμοποιήθηκαν στις ειδικές ανάγκες του Έργου.

ABSTRACT

This report deals with the design and implementation of the HYDROSCOPE database. It includes a detailed listing, grouping and classification of the various DB objects, a description of inter-object relations, and an analysis of the distributed operation, its problems and their solutions. It also includes a detailed presentation of the mapping of data, as physical entities, to the various DB objects, and of the way the issues of security, accounting and tool adaptation have been met.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1 ΓΕΝΙΚΑ.....	1
1.1 Εισαγωγή.....	1
1.2 Κύρια χαρακτηριστικά.....	2
1.3 Κοινό.....	3
1.4 Ιστορικό και ευχαριστίες.....	3
2 ΓΕΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	6
2.1 Κατάταξη πληροφοριών.....	6
2.2 Μορφές Αποθήκευσης Χρονοσειρών.....	13
2.3 Κατανεμημένη λειτουργία.....	14
3 ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΥΛΟΠΟΙΗΣΗΣ.....	18
3.1 Τοπική βάση δεδομένων.....	18
3.2 Κατανεμημένη βάση δεδομένων.....	18
3.3 Μορφή ΚΒΔ.....	20
3.4 Ταυτοποίηση αντικειμένων.....	21
3.5 Πίνακες αναφοράς και ονομασίες.....	25
3.6 Τύποι αντικειμένων και ΚΒΔ.....	26
3.7 Αποθήκευση χρονοσειρών.....	29
3.8 Αποθήκευση σταθερών.....	42
3.9 Είδη μεταβλητών και οργάνων.....	49
4 ΥΠΟΣΥΣΤΗΜΑΤΑ ΑΣΦΑΛΕΙΑΣ ΚΑΙ ΧΡΕΩΣΗΣ.....	73
4.1 Γενικά για το υποσύστημα ασφάλειας.....	73
4.2 Κατηγορίες αντικειμένων και δικαιωμάτων πρόσβασης.....	74
4.3 Κατηγορίες ομάδων χρηστών.....	76
4.4 Κωδικοί χρήστη.....	80
4.5 Πίνακες ΒΔ για το υποσύστημα ασφάλειας.....	82
4.6 Ασφάλεια εφαρμογών.....	83
4.7 Γενικά για το υποσύστημα χρέωσης.....	86
4.8 Πίνακες και διαδικασίες ΒΔ για το υποσύστημα χρέωσης.....	87
4.9 Χρέωση εφαρμογών.....	90
5 ΥΠΟΣΤΗΡΙΞΗ ΕΦΑΡΜΟΓΩΝ.....	91
5.1 Συναρτήσεις BeforeQuery () και AfterQuery ().....	91
5.2 Πρόσβαση στα δεδομένα από τις εφαρμογές.....	94
5.3 Διαχειριστικές πληροφορίες.....	99
6 ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΟΡΙΖΟΜΕΝΟΙ ΑΠΟ ΤΟ ΧΡΗΣΤΗ.....	101

6.1 Εισαγωγή.....	101
6.2 Τύπος δεδομένων “Συντεταγμένη”	101
6.3 Τύπος δεδομένων “Διατεταγμένο ζεύγος”	102
6.4 Τύπος δεδομένων “Ημερομηνία ΥΔΡΟΣΚΟΠΙΟΥ”	104
6.5 Τύπος δεδομένων “Κωδικός ΥΔΡΟΣΚΟΠΙΟΥ”	106
6.6 Περιορισμοί χρήσης συναρτήσεων οριζόμενων από το χρήστη.....	107
6.7 Συνάρτηση ελέγχου τύπου.....	108
6.8 Συναρτήσεις λέξης κατάστασης.....	109
7 ΑΝΤΙΚΕΙΜΕΝΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	112
7.1 Εισαγωγή.....	112
7.2 Πίνακες αναφοράς.....	112
7.3 Πίνακες ασφάλειας - χρέωσης - συστημάτων.....	119
7.4 Πίνακες εφαρμογών.....	123
7.5 Πίνακες αποθήκευσης σταθερών.....	131
7.6 Πίνακες αποθήκευσης πρωτογενών δεδομένων χρονοσειρών.....	134
7.7 Πίνακες αποθήκευσης δευτερογενών δεδομένων χρονοσειρών.....	136
7.8 Πίνακες αποθήκευσης ειδικών δεδομένων χρονοσειρών.....	138
7.9 Πίνακες σταδμών.....	140
7.10 Πίνακες οργάνων.....	146
7.11 Πίνακες χαρακτηριστικών χρονοσειρών.....	147
7.12 Πίνακες χαρακτηριστικών σταθερών.....	150
7.13 Πίνακες γεγονότων.....	152
7.14 Κανόνες ΒΔ.....	154
7.15 Διαδικασίες ΒΔ.....	156
7.16 Γεγονότα ΒΔ.....	157
8 ΕΠΙΠΛΕΟΝ ΣΤΟΙΧΕΙΑ.....	158
8.1 Δαίμονας αντιγραφής hrpld.....	158
8.2 Δαίμονας τροποποίησης χαρακτηριστικών χρονοσειρών ht sd.....	159
8.3 Υποστήριξη.....	160
ΑΝΑΦΟΡΕΣ.....	164
ΠΑΡΑΡΤΗΜΑ Α: ΓΛΩΣΣΑΡΙ.....	167
ΠΑΡΑΡΤΗΜΑ Β: ΠΙΝΑΚΕΣ ΑΝΑΦΟΡΑΣ.....	175
ΠΑΡΑΡΤΗΜΑ Γ: ΚΑΝΟΝΕΣ ΚΑΙ ΔΙΑΔΙΚΑΣΙΕΣ ΒΔ.....	211
ΠΑΡΑΡΤΗΜΑ Δ: ΔΙΑΔΙΚΑΣΙΑ ΕΓΚΑΤΑΣΤΑΣΗΣ.....	222

1 ΓΕΝΙΚΑ

1.1 Εισαγωγή

Η παρούσα μελέτη πραγματοποιείται τη μορφή της ΒΔ¹ { Γ } της Εθνικής Τράπεζας Υδρολογικής και Μετεωρολογικής Πληροφορίας που αναπτύσσεται στα πλαίσια του Ερευνητικού Έργου ΥΔΡΟΣΚΟΠΙΟ (STRIDE Hellas) [Το193]². Αφορά στην εργασία 12 του σταδίου “Γενική Ανάλυση” του Έργου με τίτλο “Τελική Σχεδίαση Συστήματος”. Στόχος αυτής της εργασίας ήταν ο προσδιορισμός των λειτουργικών παραμέτρων και χαρακτηριστικών και ο σχεδιασμός των διάφορων τμημάτων της ΒΔ. Το τελικό αποτέλεσμα της μελέτης είναι ο καθορισμός και η περιγραφή όλων των λειτουργικών στοιχείων της ΒΔ: πίνακες, όψεις, κλειδιά, κανόνες, διαδικασίες ΒΔ, βοηθητικά προγράμματα κοκ.

Οι στόχοι του σχεδιασμού είναι (χωρίς σειρά προτεραιότητας):

1.1.1 Συνολική αποθήκευση. Η ΒΔ να είναι σε θέση να αποθηκεύσει το σύνολο των υδρολογικών, υδρογεωλογικών και μετεωρολογικών δεδομένων της Τράπεζας, χωρίς εξαιρέσεις ή περιορισμούς.

1.1.2 Ευελιξία. Η ΒΔ να είναι δυνατό να ανταποκριθεί χωρίς μείωση της λειτουργικότητας της στις ποικίλες απαιτήσεις αποθήκευσης και επεξεργασίας στοιχείων των διάφορων συμμετεχόντων Φορέων. Επίσης να είναι σε θέση να ενσωματώσει τυχόν νέες απαιτήσεις ή τεχνικές χωρίς σημαντικές αλλαγές. Αυτό το χαρακτηριστικό οδηγεί τόσο στην ευρύτερη χρήση του συστήματος όσο και στην μακροζωία του.

1.1.3 Συμμόρφωση με τις προδιαγραφές και τις απαιτήσεις χρήσης. Ο σχεδιασμός της ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ πρέπει να συμφωνεί με τις απαιτήσεις χρήσης του όπως αυτές εκφράστηκαν κατά τη φάση της ανάλυσης του έργου.

1.1.4 Διατήρηση διαχειριστικής αυτονομίας Φορέων. Η διατήρηση της διαχειριστικής αυτονομίας των συμμετεχόντων Φορέων μέσα από ένα αποτελεσματικό

1 Οι επεξηγήσεις των συντομογραφιών, των όρων και των ελληνικών μεταφράσεων των αγγλικών όρων βρίσκονται στο Γλωσσάρι (Παράρτημα Α). Ένας όρος που ακολουθείται από Γ σε αγκύλες, { Γ }, περιέχεται και επεξηγείται στο Γλωσσάρι.

2 Στις βιβλιογραφικές αναφορές ακολουθείται ο συνήθης συμβολισμός των άρθρων - βιβλίων Πληροφορικής: [xxxYYz] όπου xxx είναι κάποιο χαρακτηριστικό 3 ή περισσότερων χαρακτήρων για το όνομα του συγγραφέα, YY είναι η χρονολογία έκδοσης και z προαιρετικό γράμμα a, b, c, ... ή α, β, γ, ... για να διακρίνονται άρθρα του ίδιου συγγραφέα και της ίδιας χρονολογίας. Για παράδειγμα, [Το193], [INGRES91a], [Πιπ92b], [Codd70] κοκ

σύστημα ασφάλειας και κατανεμημένης λειτουργίας είναι ένας από τους βασικούς στόχους της σχεδίασης.

1.1.5 Βέλτιστη χρήση υπάρχοντος εξοπλισμού. Ο διαθέσιμος εξοπλισμός σε σταθμούς εργασίας, προσωπικούς υπολογιστές και δικτυακές συσκευές θα πρέπει να αξιοποιηθεί πλήρως και αποτελεσματικά, χωρίς να υποστεί κορεσμό. Στο στόχο αυτό εντάσσεται τόσο η ταχύτητα λειτουργίας όσο και η οικονομία αποθήκευσης και κυκλοφορίας πληροφοριών πάνω από το δίκτυο.

1.1.6 Ανοχή σε σφάλματα. Δυνατότητα του συστήματος να αντιμετωπίζει με επιτυχία τυχόν σφάλματα, κυρίως στη λειτουργία του δικτύου.

1.1.7 Εξελιγμένη αλλά και ώριμη τεχνολογία. Η χρήση τεχνολογίας ΒΔ αρκετά εξελιγμένης ώστε να ενσωματώνει νέες εξελίξεις θετικές για την πληρότητα και λειτουργικότητα της ΒΔ αλλά και αρκετά ώριμης και δοκιμασμένης ώστε να μη δέτει σε κίνδυνο την επιτυχία του Ερευνητικού Έργου είναι ένας από τους βασικούς στόχους.

1.1.8 Λειτουργικότητα και ευκολία χρήσης. Η ΒΔ θα πρέπει να είναι εύκολη στη χρήση και στην κατανόηση της, όσο εύκολο μπορεί βέβαια να είναι ένα σύστημα λογισμικού αυτού του μεγέθους.

1.1.9 Υποστήριξη Ελληνικών. Η ΒΔ θα πρέπει να υποστηρίζει Ελληνικά στην αποθήκευση των δεδομένων της και στην επικοινωνία της με τις εφαρμογές -που θα πρέπει και αυτές με τη σειρά τους να υποστηρίζουν Ελληνικά.

Με βάση αυτούς τους στόχους αναπτύχθηκε η ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ. Η ανάπτυξη έγινε σταδιακά, με τη μέθοδο του πρωτότυπου, όπου ένα αρχικό σχήμα υποβαλλόταν κάθε φορά σε δοκιμές, εμπλουτιζόταν και διορθωνόταν, για να αποτελέσει τη βάση για μια επόμενη, καλύτερη προσέγγιση.

1.2 Κύρια χαρακτηριστικά

Τα κύρια χαρακτηριστικά της ΒΔ είναι τα ακόλουθα:

1.2.1 Κατανεμημένη. Σε κάθε κόμβο υπάρχει μια ΤΒΔ { Γ } η οποία αποθηκεύει τις πληροφορίες του κόμβου. Η λογική ένωση των ΤΒΔ αποτελεί την ΚΒΔ { Γ }. Οι ΤΒΔ και η ΚΒΔ έχουν το ίδιο σχήμα { Γ } ώστε να διευκολύνεται η ανάπτυξη των εφαρμογών και η προσπέλαση από τους χρήστες. Επιπλέον, η ΚΒΔ παρέχει διαφάνεια τοποθέτησης { Γ } ως προς τα δεδομένα της, εκτός δηλαδή από την ταχύτητα πρόσβασης δεν υπάρχει για το χρήστη (και τον προγραμματιστή ως ένα -σημαντικό- βαθμό) τρόπος να αντιληφθεί την ακριβή προέλευση των δεδομένων.

1.2.2 Σχεσιακή. Ακολουθείται το σχεσιακό μοντέλο [Codd70] που αυτή τη στιγμή είναι ο κυρίαρχος τρόπος απεικόνισης της πραγματικότητας σε μια ΒΔ. Επίσης είναι αυτό το

μοντέλο που δίνει τις περισσότερες επιλογές ως προς τον εξυπηρετητή { Γ } και τα εργαλεία ανάπτυξης που θα χρησιμοποιηθούν και μάλιστα με χρήση των πλέον εξελιγμένων τεχνολογιών σήμερα.

1.2.3 Πολυεπίπεδη. Τα δεδομένα αποθηκεύονται σε πολλαπλά επίπεδα, τόσο ως προς το βαθμό επεξεργασίας όσο και ως προς τη χρονική εξέλιξη τους.

1.2.4 Απλότητα. Το σχήμα της ΒΔ είναι αρκετά απλό, παρ' ό,τι επεκτείνεται σε μεγάλο εύρος. Οι οντότητες της έχουν στενή σχέση με αυτές του πραγματικού κόσμου και είναι εύκολα αναγνωρίσιμες από τους χρήστες της.

1.2.5 Μειωμένη χρήση δικτύου. Καταβάλλεται σημαντική προσπάθεια ώστε η χρήση του δικτύου, που είναι το πλέον αργό από τα διαθέσιμα μέσα αποθήκευσης και διακίνησης πληροφοριών (κεντρική μνήμη, σκληρός δίσκος, δίκτυο) να είναι περιορισμένη ώστε η γενικότερη απόκριση του συστήματος να μην εξαρτάται τόσο πολύ από την ύπαρξη του.

1.3 Κοινό

Η απρόσκοπτη ανάγνωση και πλήρης κατανόηση της παρούσας μελέτης προϋποθέτει καταρχήν υπομονή αλλά και ορισμένες βασικές γνώσεις Πληροφορικής, Υδρολογίας και Μετεωρολογίας. Κυρίως απευθύνεται στους προγραμματιστές εφαρμογών του ΥΔΡΟΣΚΟΠΙΟΥ αλλά και σε όσους επιθυμούν να γνωρίσουν σε βάθος το υπόβαθρο του Έργου, τη βάση δεδομένων. Η φιλοσοφία και τα κίνητρα πίσω από τις επιλογές που έγιναν πρέπει να είναι σαφή στον προσεκτικό αναγνώστη. Έχει καταβληθεί προσπάθεια να παρουσιασθεί όχι μόνο το "τι" αλλά και το "πώς" και το "γιατί".

Ο λιγότερο ειδικευμένος χρήστης μπορεί να αρκεστεί στα τρία πρώτα κεφάλαια για να πάρει μια καλή ιδέα της μορφής της βάσης και να μπορεί ευκολότερα να κατανοήσει τη λειτουργία των εφαρμογών και του όλου συστήματος. Τα υπόλοιπα κεφάλαια έχουν ενδιαφέρον για κάποιον που ζητά συγκεκριμένες πληροφορίες.

1.4 Ιστορικό και ευχαριστίες

Η μελέτη αυτή έχει ως αντικείμενο την παρουσίαση της τελικής μορφής της ΒΔ του συστήματος του ΥΔΡΟΣΚΟΠΙΟΥ. Η μορφή αυτή ήταν το αποτέλεσμα μιας επίπονης και μακράς εργασίας κατά τη διάρκεια της οποίας έπρεπε να υπερπηδηθούν σημαντικά εμπόδια. Χαρακτηριστικά αναφέρονται:

- Η έλλειψη ακριβούς γνώσης μερικών εμπλεκόμενων - συνεργαζόμενων επιστημόνων και ερευνητών από διάφορους συμμετέχοντες Φορείς για την επίθυμη και προδιαγραμμένη τελική μορφή του συστήματος: μια άμεσα προσπελάσιμη Κατανεμημένη ΒΔ, ενιαία για το σύνολο των υδρολογικών, υδρογεωλογικών και

μετεωρολογικών πληροφοριών. Συνακόλουθα ήταν πρόβλημα και η έλλειψη γνώσης των πολλών επιπλέον δυνατοτήτων του συστήματος σε σχέση με υπάρχοντα χειρογραφικά ή και αυτόματα παλαιότερης τεχνολογίας συστήματα, αλλά και γενικότερα η έλλειψη γνώσης και ενημέρωσης για τις δυνατότητες της σύγχρονης τεχνολογίας, όπως για παράδειγμα: η δυνατότητα ταυτόχρονης πρόσβασης στο σύνολο των δεδομένων και μάλιστα με διαφάνεια τοποθέτησης και χωρίς άμεση πρόσβαση σε απομακρυσμένα υπολογιστικά συστήματα, η δυνατότητα (αλλά και η υποχρέωση) εύκολης και άμεσης επιλογής του υποσυνόλου δεδομένων που ζητείται, αντί για ανάκτηση όλων των δεδομένων και εκ των υστέρων αποκλεισμού των μη ενδιαφερόντων από αυτά, η δυνατότητα λειτουργίας σε γραφικό περιβάλλον, η ύπαρξη πολλαπλών επιπέδων ασφάλειας³ κ.ο.κ.

- Σε πολλά σημεία το αποτέλεσμα διαφόρων εργασιών Γενικής Ανάλυσης που προηγήθηκαν του τελικού σχεδιασμού του συστήματος δεν ήταν το αναμενόμενο, από την άποψη ότι δεν έδωσε μια σαφή εικόνα των απαιτήσεων και των λειτουργικών προδιαγραφών και δεν ήταν δυνατό να αποτελέσει ασφαλή οδηγό για τον προβληματίστο σχεδιασμό.
- Σε σχέση με την πολυπλοκότητα, την πολυπαραμετρικότητα και την ευρύτητα του προβλήματος ο διαθέσιμος χρόνος για το σχεδιασμό του συστήματος ήταν περιορισμένος, δεδομένου μάλιστα πως ταυτόχρονα σχεδόν θα έπρεπε να αναπτύσσονται και εφαρμογές.
- Δεν υπήρχαν διαθέσιμα εργαλεία υποβοήθησης του σχεδιασμού της ΒΔ αλλά και ούτε ο απαραίτητος χρόνος για εκμάθηση και παραγωγική χρησιμοποίησή τους.

Η μεθοδολογία που χρησιμοποιήθηκε προκειμένου να ξεπεραστούν τα παραπάνω και τυχόν άλλα προβλήματα και να πάρει την τελική του μορφή το σχήμα { Γ } της ΒΔ ήταν η ακόλουθη:

- η ΒΔ αναπτύχθηκε καταρχήν σε μορφή πρωτότυπου { Γ } με κύρια έμφαση στις διαχειριστικές πληροφορίες⁴ (σταθμούς, όργανα κτλ.) και με σχετικά στοιχειώδη υποστήριξη αποθήκευσης δεδομένων με σκοπό την ανάπτυξη των πρώτων εφαρμογών (υδρομετεωρολογικός χάρτης, μητρώο σταθμών) και την εξαγωγή συμπερασμάτων σχετικά με τις απαιτούμενες δυνατότητες και τα προβλήματα.

³ Εννοείται βέβαια πως κανένα σύστημα ΒΔ δεν είναι ασφαλέστερο από ένα σύνολο χάρτινων ή ακόμα και μαγνητικών αρχείων κλειδωμένων σε κάποιο υπόγειο. Αυτή όμως η μορφή ασφάλειας δεν συμβαδίζει ούτε με τη σύγχρονη τεχνολογία ούτε με την ουσία και το σκοπό του ΥΔΡΟΣΚΟΠΙΟΥ.

⁴ Οι έννοιες αυτές όπως και όσες αναφέρονται στη συνέχεια αυτής της παραγράφου επεξηγούνται φυσικά παρακάτω

- αναπτύχθηκαν διάφορα εργαλεία και υποσυστήματα λογισμικού απαραίτητα -όπως ήταν σαφές από την αρχική φάση- για το σχεδιασμό (τύποι δεδομένων οριζόμενοι από το χρήστη, κριτήρια επιλογής κοκ).
- το πρωτότυπο ελέγχθηκε και πέρασε από διαδοχικές αναθεωρήσεις και επεκτάσεις με σκοπό τόσο την αύξηση της λειτουργικότητάς του και την ενσωμάτωση νέων δυνατοτήτων (αποθήκευση πρωτογενών και δευτερογενών χρονοσειρών δεδομένων, αποθήκευση σταθερών δεδομένων, γεγονότα, επιπλέον δεδομένα κοκ) όσο και την εξάλειψη λαθών και προβλημάτων.
- όταν η μορφή της ΒΔ έφθασε σε μια σχετικά σταθερή κατάσταση, κατασκευάστηκαν επιπλέον υποσυστήματα απαραίτητα για την κανονική της λειτουργία (δαίμονες αντιγραφής και χρονοσειρών, ασφάλεια και χρέωση, εγκατάσταση).
- σε όλη τη διάρκεια αυτής της διαδικασίας ορισμένες τεχνικές αναφορές του ΥΔΡΟΣΚΟΠΙΟΥ, που ήταν δυνατό να αξιοποιηθούν, χρησιμοποιήθηκαν ως βασική αναφορά. Το κύριο βάρος όμως της συλλογής πληροφοριών για τον ακριβή καθορισμό των λειτουργικών προδιαγραφών του συστήματος έπεσε σε προσωπικές συνεντεύξεις με ερευνητές του Έργου σχετικά με το αντικείμενο της αρμοδιότητάς τους, ώστε να καθορισθούν σαφώς και επακριβώς οι απαιτήσεις τους από τη ΒΔ. Η διαδικασία ήταν αμφίδρομη και μακρόχρονη, με υποβολή ερωτήσεων και εποικοδομητική συζήτηση ώστε να βρεθούν λύσεις σε όλα τα ζητήματα που προέκυψαν. Η προσωπική επαφή έπαιξε έτσι σημαντικότερο ρόλο από την έγγραφη και "επίσημη" τεκμηρίωση.

Πολλοί άνθρωποι συνέβαλλαν αποφασιστικά με τις προτάσεις, τις παρατηρήσεις, τα σχόλια, την κριτική, την εργασία ή και απλά τη συμπαράστασή τους στην τελική μορφή του συστήματος. Αυτοί είναι οι: Δημήτρης Κουτσογιάννης, Γιάννης Ναλμπάντης, Νίκος Μαμάσης, Γιώργος Τσακαλίας, Κάτια Πιπιλή, Αντώνης Χριστοφίδης, Αλέξανδρος Μανέτας, Γιάννα Σταματάκη, Δημήτρης Καλογεράς, Δώρα Τζεφέρη, Ανδρέας Σακελλαρίου, Πάνος Παπανικολάου, Γιάννης Παπαγεωργίου, Γιάννης Μαλλάς, Νίκος Δρης, Δημήτρης Τολίκας, Νικόλας Γεωργιάδης, Σταύρος Λαδάς, Μάκης Τηλιγάδας, Τίμος Σελλής και Γιώργος Γαζέπης. Σε όλους οφείλονται θερμότατες ευχαριστίες.

2 ΓΕΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

2.1 Κατάταξη πληροφοριών

Οι πληροφορίες στη ΒΔ μπορούν να καταταχθούν σε τρεις κύριες κατηγορίες:

2.1.1 Πληροφορίες εφαρμογών και βοηθητικές πληροφορίες. Στην κατηγορία αυτή εντάσσονται οι πληροφορίες της ΒΔ οι οποίες σχετίζονται με τις εφαρμογές και διάφορα άλλα υποσυστήματα του ΥΔΡΟΣΚΟΠΙΟΥ, πχ. πληροφορίες για διάφορα μηνύματα, για χάρτες, για κείμενα βοήθειας κοκ. Επίσης στην κατηγορία αυτή εντάσσονται οι πίνακες αναφοράς { Γ }. Τέλος υπάρχουν και πληροφορίες για τα υποσυστήματα ασφάλειας και χρέωσης. Γενικά οι πίνακες αυτής της κατηγορίας είναι στατικοί, δεν αλλάζουν δηλαδή με την πάροδο του χρόνου (εξαιρούνται οι πίνακες ασφάλειας και χρέωσης). Δεν έχουν άμεση σχέση με τα αποθηκευμένα δεδομένα αλλά με την ορδή λειτουργία των εφαρμογών και γενικότερα του συστήματος.

2.1.2 Διαχειριστικές πληροφορίες. Οι διαχειριστικές πληροφορίες ή μεταδεδομένα { Γ } είναι οι πληροφορίες μέσω των οποίων γίνεται η πρόσβαση στα πραγματικά δεδομένα της ΒΔ. Διαχειριστικές πληροφορίες είναι:

2.1.2.1 Σταθμοί. Οι μετρητικοί σταθμοί είναι η αρχική μονάδα επεξεργασίας της πληροφορίας. Κάθε σταθμός είναι μια αυτόνομη οντότητα που ανήκει σε μια μόνο υπηρεσία (Φορέα). Έχει καθορισμένη γεωγραφική θέση (αν και στη διάρκεια της ζωής του μπορεί να αλλάξει ελαφρά η θέση του ή το υψόμετρο του) και καθορισμένα χαρακτηριστικά. Κάθε σταθμός περιέχει όργανα μετρήσεων (βλ. παρακάτω).

Οι σταθμοί χωρίζονται σε δύο κατηγορίες:

2.1.2.1.1 Πρωτεύοντες σταθμοί. Οι πρωτεύοντες σταθμοί (ή σταθμοί εθνικού δικτύου) { Γ } είναι οι βασικοί σταθμοί του ΥΔΡΟΣΚΟΠΙΟΥ. Συντηρούνται και διατηρούνται από τις αρμόδιες υπηρεσίες σε κανονική επιχειρησιακή λειτουργία, τα όργανα τους μετρούν κανονικά και τακτικά και γενικά είναι "ενεργοί". Ο αριθμός τους είναι περίπου 2000.

2.1.2.1.2 Δευτερεύοντες σταθμοί. Οι δευτερεύοντες σταθμοί δεν είναι γενικά τόσο σημαντικοί, μια και δεν έχουν κανονική λειτουργία. Μπορεί να λειτουργούσαν στο παρελθόν και στη συνέχεια έγιναν "ανεργοί". Πχ. μια γεώτρηση της οποίας τα χαρακτηριστικά μετρήθηκαν μια φορά μόνο, κατά την έναρξη της λειτουργίας της, μπορεί -κατά περίπτωση- να θεωρηθεί δευτερεύων

σταθμός. Ο αριθμός των δευτερευόντων σταθμών ξεπερνά τις 100000 και τα στοιχεία τους κατά τεκμήριο χρησιμοποιούνται σπάνια. Σε περίπτωση που ένας σταθμός είναι δευτερεύων, τότε όλες οι διαχειριστικές του πληροφορίες, όπως θα ορισθούν στη συνέχεια (όργανα, χαρακτηριστικά χρονοσειρών και σταθερών, γεγονότα), είναι και αυτές δευτερεύουσες. Και γενικά, Αν ένας σταθμός ανήκει σε μια κατηγορία, τότε και όλες οι διαχειριστικές πληροφορίες του ανήκουν στην ίδια κατηγορία.

2.1.2.2 Όργανα. Κάθε σταθμός περιέχει μετρητικά όργανα. Κάθε μετρητικό όργανο ανήκει σε έναν μόνο σταθμό -και κατά συνέπεια σε μία μόνο υπηρεσία-, εκτελεί περιοδικές (με σταθερή ή κατά περιόδους μεταβλητή περίοδο) ή ακανόνιστες μετρήσεις μιας υδρολογικής, υδρογεωλογικής ή μετεωρολογικής μεταβλητής. Η χρονική διακριτότητα $\{ \Gamma \}$, δηλαδή η περίοδος μέτρησης των οργάνων -αν υπάρχει- μπορεί να είναι οποιαδήποτε. Στο ΥΔΡΟΣΚΟΠΙΟ είναι δυνατές οποιοσδήποτε χρονικές διακριτότητες με μέγιστη το ένα πρώτο λεπτό. Σε κάθε περίπτωση τα όργανα παράγουν δεδομένα χρονοσειρών. Χωρίζονται με τη σειρά τους σε δύο κατηγορίες:

2.1.2.2.1 Πραγματικά όργανα. Τα πραγματικά όργανα είναι αυτά που εκτελούν απευθείας μετρήσεις σε διάφορες μεταβλητές. Οι μετρήσεις αυτές μπορεί να προέρχονται είτε από μια μετρητική συσκευή (πχ. θερμόμετρο, βροχόμετρο) είτε από απλή ανθρώπινη παρατήρηση (πχ. νέφωση, ορατότητα).

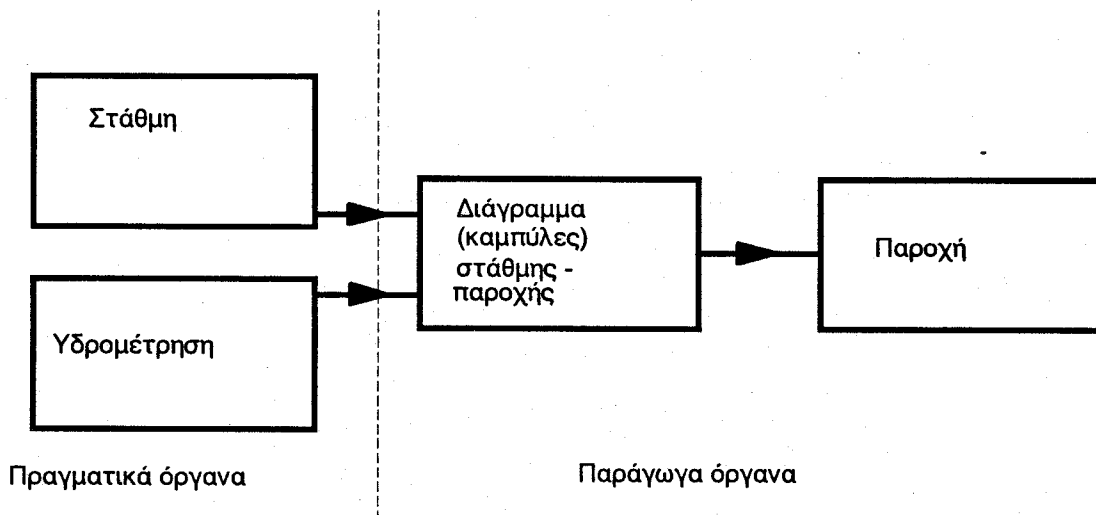
2.1.2.2.2 Παράγωγα όργανα. Τα παράγωγα $\{ \Gamma \}$ όργανα είναι αυτά τα οποία των οποίων οι "μετρήσεις" είναι το αποτέλεσμα υπολογισμών και επεξεργασίας πάνω σε πρωτογενή ή δευτερογενή δεδομένα (βλ. παρακάτω) ενός ή περισσότερων άλλων οργάνων (πραγματικών ή παράγωγων) [Λιακ92]. Οι χρονικές διακριτότητες των παράγωγων οργάνων πρέπει να είναι μικρότερες ή ίσες από των αρχικών⁵, δηλαδή αρχικά όργανα διακριτότητας μιας ημέρας, που παράγουν δηλαδή μετρήσεις κάθε μία ημέρα, μπορούν να έχουν παράγωγα όργανα ημερήσιας ή μικρότερης, πχ. μηνιαίας, ετήσιας διακριτότητας. Παραδείγματα παράγωγων οργάνων είναι η εξατμισοδιαπνοή και η παροχή. Η τελευταία μάλιστα προέρχεται από ένα διάγραμμα στάθμης - παροχής, το οποίο είναι επίσης παράγωγο όργανο, το οποίο προέρχεται από πραγματικά όργανα υδρομετρήσεων και στάθμης.

2.1.2.3 Χαρακτηριστικά χρονοσειρών. Κάθε χρονοσειρά που παράγεται από ένα όργανο έχει ορισμένα χαρακτηριστικά όπως τη διακριτότητα της (πχ. ωριαία, ημερήσια, μηνιαία, ετήσια κτλ.), το είδος της (πχ. πρωτογενής, δευτερογενής -βλ.

5

Μικρότερη διακριτότητα σημαίνει βέβαια μεγαλύτερο διάστημα ανάμεσα σε δύο μετρήσεις. Έτσι, οι μηνιαίες μετρήσεις έχουν μικρότερη χρονική διακριτότητα από τις ημερήσιες.

παρακάτω), το αν χρησιμοποιείται γραμμική αποθήκευση (επίσης βλ. παρακάτω), την ημερομηνία αρχής και την ημερομηνία τέλους. Τυχόν απουσία εγγραφής χρονοσειράς με αυτά τα χαρακτηριστικά σημαίνει αυτόματα πως δεν υπάρχουν τα αντίστοιχα δεδομένα. Κανονικά βέβαια οι χρονοσειρές (ως "φυσική" έννοια) έχουν αρχή την ημερομηνία πρώτης λειτουργίας του οργάνου και τέλος το τέλος λειτουργίας. Εχουν όμως και χρονικά διαστήματα στα οποία δεν υπάρχουν δεδομένα για διάφορους λόγους (πχ. βλάβη οργάνου), δηλαδή "κενά." Για κάθε τέτοιο διάστημα ακολουθείται η εξής τακτική: υπάρχουν δύο εγγραφές χαρακτηριστικών χρονοσειρών



Σχήμα 2.1.1: Σχέση πραγματικών - παράγωγων οργάνων και παράδειγμα: η παροχή ποταμού

για το συγκεκριμένο όργανο, μία με ημερομηνία λήξης την τελευταία ημερομηνία για την οποία υπάρχουν δεδομένα χρονοσειράς πριν το κενό και μία με ημερομηνία αρχής την πρώτη ημερομηνία για την οποία υπάρχουν δεδομένα μετά το κενό. Όλα τα άλλα χαρακτηριστικά είναι ίδια ανάμεσα στις δύο εγγραφές χαρακτηριστικών χρονοσειρών -σε περίπτωση βέβαια που έχει αλλάξει κάτι (πχ. μορφή αποθήκευσης) αυτό το γεγονός καταγράφεται με αλλαγή του αντίστοιχου χαρακτηριστικού. Έτσι, τα διαστήματα για τα οποία δεν υπάρχουν εγγραφές χαρακτηριστικών είναι αυτά για τα οποία δεν υπάρχουν δεδομένα. Συνεπώς τα χαρακτηριστικά χρονοσειρών δεν περιγράφουν μόνο τη διαθεσιμότητα των χρονοσειρών (πχ. "υπάρχει για το υπο εξέταση όργανο μια ημερήσια χρονοσειρά πρωτογενών δεδομένων") αλλά και τη διαθεσιμότητα των δεδομένων της χρονοσειράς (πχ. "υπάρχει για το υπο εξέταση όργανο μια ημερήσια χρονοσειρά πρωτογενών δεδομένων για τα χρονικά διαστήματα $\Delta 1 - \Delta 2$, $\Delta 3 - \Delta 4$ και $\Delta 5 - \Delta 6$ "). Εννοείται βέβαια πως, επειδή με κατάλληλες μεθόδους συμπλήρωσης τυχόν "κενά" στις χρονοσειρές είναι δυνατό να

“γεμίσουν” (συμπληρωθούν) υπάρχει λογισμικό που ανάλογα τροποποιεί αυτές τις εγγραφές.

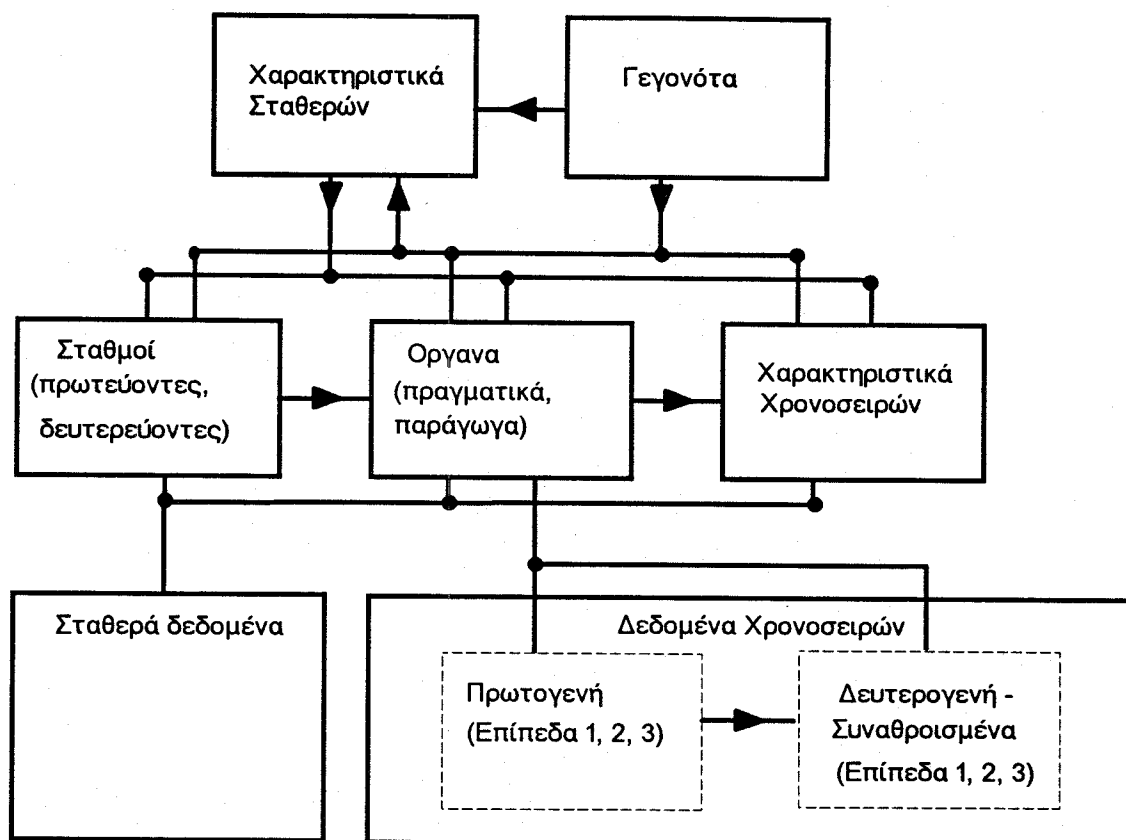
2.1.2.4 Χαρακτηριστικά σταθερών δεδομένων. Οι σταθμοί και τα όργανα μπορούν να συνοδεύονται από σταθερά δεδομένα (βλ. παρακάτω), σε αντιστοιχία με τις χρονοσειρές. Τα χαρακτηριστικά αυτών των δεδομένων, όπως το είδος, η ημερομηνία σχηματισμού κοκ. περιγράφονται όπως οι αντίστοιχες διαχειριστικές πληροφορίες.

2.1.2.5 Γεγονότα. Όλα τα προηγούμενα “αντικείμενα”, δηλαδή οι σταθμοί, τα όργανα και τα χαρακτηριστικά χρονοσειρών και σταθερών, όπως επίσης και τα ίδια τα δεδομένα μπορεί να συνοδεύονται από “γεγονότα”. Τα γεγονότα είναι πληροφορίες για οτιδήποτε σημαντικό συμβαίνει στη ζωή του αντικειμένου και καταγράφουν πλήρεις πληροφορίες που ξεκινούν από το είδος του γεγονότος και την ημερομηνία που συνέβη και φτάνουν ως μια πλήρη αναφορά για το τι συνέβη, σε κατάλληλο αρχείο του $\Lambda\Sigma\{\Gamma\}$. Παραδείγματα γεγονότων είναι η έναρξη ή η λήξη λειτουργίας ενός σταθμού ή ενός οργάνου, η εφαρμογή ενός ελέγχου ή η συμπλήρωση μιας χρονοσειράς κοκ.

Όλες οι διαχειριστικές πληροφορίες οργανώνονται κατάλληλα σε αλληλοσυνδεόμενους πίνακες και μητρώα ώστε να είναι εύκολη η πρόσβαση τους. Έτσι υπάρχει μητρώο σταθμών, οργάνων, χρονοσειρών, σταθερών και γεγονότων.

2.1.3 Δεδομένα. Η κατηγορία αυτή περιλαμβάνει τον “πυρήνα” της ΒΔ, δηλαδή τα υδρολογικά, υδρογεωλογικά και μετεωρολογικά δεδομένα που αποθηκεύονται σε αυτήν (και προσπελαύνονται μέσω των διαχειριστικών πληροφοριών). Χωρίζονται δε σε δύο υποκατηγορίες, ανάλογα με τον τύπο τους:

2.1.3.1 Σταθερά δεδομένα. Τα σταθερά (ή στατικά) δεδομένα είναι πληροφορίες σχετικές με σταθμούς ή όργανα οι οποίες χαρακτηρίζουν το αντίστοιχο αντικείμενο και είτε σχηματίζονται μια φορά και δεν αλλάζουν ποτέ είτε αλλάζουν σπάνια και ισχύουν για ένα σχετικά μεγάλο διάστημα της ζωής του αντικειμένου. Τα σταθερά δεδομένα είναι συνοδευτικά των αντικειμένων στα οποία αναφέρονται αλλά δεν είναι διαχειριστικές πληροφορίες διότι πολλές φορές αυτά είναι τα ζητούμενα. Σταθερά δεδομένα είναι η λιθολογική τομή, η περιγραφή της οπής και η περιγραφή της σωλήνωσης ενός σταθμού υδρογεωλογίας (δεν αλλάζουν ποτέ) [Γεωργ93], η διατομή ενός ποταμού και τα υδραυλικά του χαρακτηριστικά (συντελεστής Manning, κλίση) για ένα όργανο υδρομέτρησης (αλλάζει όταν αλλάζει η διατομή του ποταμού, στην ακραία



Σχήμα 2.1.2: Κατηγορίες πληροφοριών και σχέσεις μεταξύ τους

περίπτωση με κάθε υδρομέτρηση), η καμπύλη στάθμης - επιφάνειας - όγκου και η καμπύλη στάθμης - παροχής υπερχειλιστή ενός ταμιευτήρα που αναφέρονται σε έναν σταθμό στον ταμιευτήρα (δεν αλλάζουν ποτέ) και τέλος, για ένα παράγωγο όργανο, ο κατάλογος των άλλων οργάνων από τα οποία υπολογίζονται οι τιμές του παράγωγου (δεν αλλάζει ποτέ). Τα σταθερά δεδομένα έχουν επίσης το χαρακτηριστικό πως δεν υπόκεινται σε οποιαδήποτε περαιτέρω επεξεργασία, όπως συμβαίνει με τα δεδομένα χρονοσειρών: απλά σχηματίζονται και αυτούσια αποθηκεύονται στη ΒΔ.

2.1.3.2 Χρονοσειρές. Είναι τα δεδομένα που, άμεσα ή έμμεσα, παράγονται από τα πραγματικά ή παράγωγα μετρητικά όργανα, είτε αυτά μετρούν περιοδικά είτε ακανόνιστα [Σακ92]. Κάθε εγγραφή χρονοσειράς αναφέρεται σε κάποιο πραγματικό ή παράγωγο όργανο. Χωρίζονται σε δύο υποκατηγορίες:

2.1.3.2.1 Πρωτογενή. Είναι τα δεδομένα που άμεσα παράγονται από τα όργανα, είτε με μέτρηση (πραγματικά όργανα) είτε με υπολογισμό (παράγωγα όργανα). Αποθηκεύονται στη ΒΔ στην ίδια χρονική διακρίσιμότητα (περίοδο μετρήσεων ή χρονικό βήμα) με αυτή του οργάνου και στις ημερομηνίες που σημειώθηκαν ή στις οποίες αναφέρονται οι μετρήσεις [Τσιμ92]. Έτσι μπορεί να

θεωρηθεί πως τα πρωτογενή δεδομένα έχουν πάντα την ίδια χρονική διακριτότητα με αυτή του οργάνου. Τα παράγωγα όργανα δίνουν και αυτά πρωτογενή δεδομένα, τα οποία μπορούν να συναδροισθούν σε δευτερογενή (βλ. αμέσως παρακάτω). Καταχρηστικά θα μπορούσε να ορισθεί και η υποκατηγορία των παράγωγων δεδομένων ως ισοδύναμη με αυτή των πρωτογενών, διάκριση που μάλλον σύγχυση θα επέφερε. Πρέπει πάντως να γίνει σαφές πως οι τιμές των παράγωγων οργάνων μπορούν να παράγονται τόσο από πρωτογενή όσο και από δευτερογενή δεδομένα άλλων οργάνων. Πχ. ένα θερμόμετρο δημιουργεί μια ημερήσια χρονοσειρά που συναδροίζεται σε μια μηνιαία δευτερογενή χρονοσειρά, η οποία μπορεί να χρησιμοποιηθεί για τον υπολογισμό μιας μηνιαίας (ή ετήσιας...) χρονοσειράς εξατμισοδιαπνοής (παράγωγο όργανο).

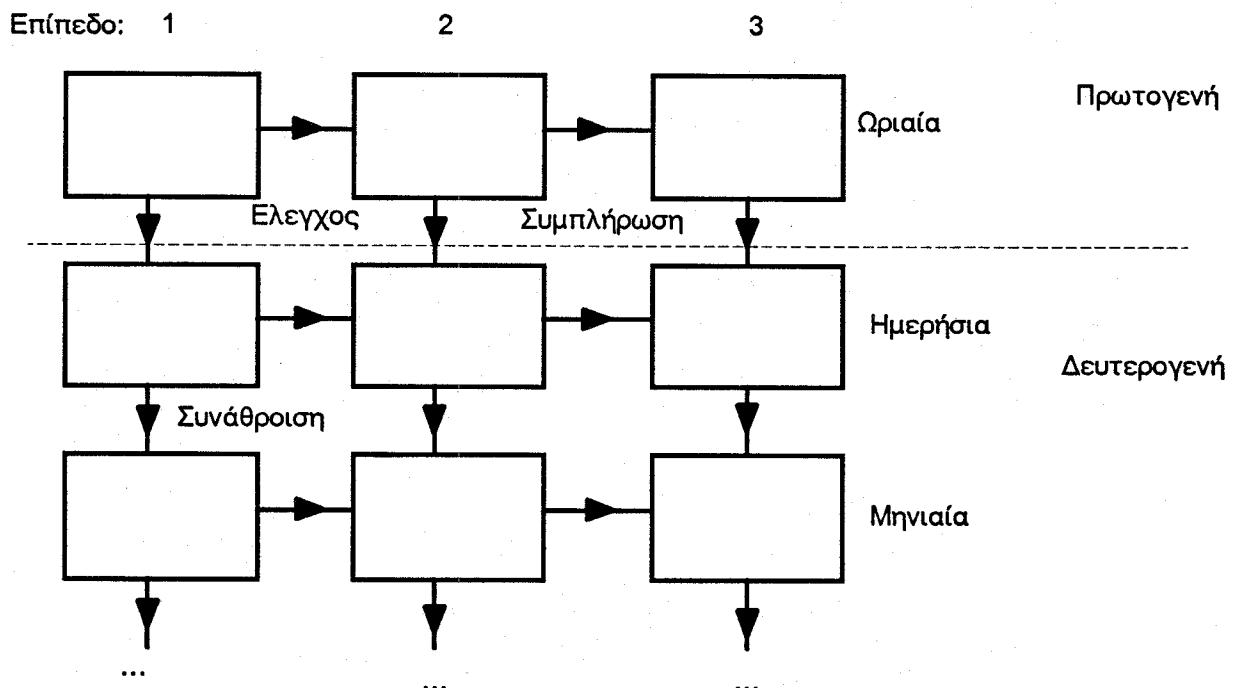
Τα πρωτογενή δεδομένα ταξινομούνται σε τρία επίπεδα, τα 1, 2 και 3 [Ναλ93]. Στο επίπεδο 1 αποθηκεύονται τα δεδομένα όπως προκύπτουν από τις μετρήσεις ή τους υπολογισμούς. Στο επίπεδο 2 υπάρχουν τα πρωτογενή δεδομένα (ίδιας χρονικής διακριτότητας με τις αρχικές μετρήσεις) επιπέδου 1 στα οποία έχουν (σε κάθε μεμονωμένη τιμή) εφαρμοσθεί διάφοροι έλεγχοι (χωρικής, εσωτερικής και χρονικής συνέπειας, ακραίων τιμών κοκ), έχουν συμπληρωθεί μεμονωμένες ελλείπουσες τιμές και έχουν χαρακτηριστεί οι τιμές ως προς την αξιοπιστία τους. Στο επίπεδο 3 έχουν εφαρμοσθεί στα δεδομένα του επιπέδου 2 εκτεταμένες διαδικασίες συμπλήρωσης και ομογενοποίησης. Με κατάλληλη αποθήκευση μέσω της λέξης κατάστασης (βλ. παρακάτω) δεν απαιτείται επανάληψη όλου του όγκου των πρωτογενών δεδομένων για αποθήκευση και των 3 επιπέδων αλλά μόνο των τιμών στις οποίες το ένα επίπεδο διαφέρει από το προηγούμενο του.

2.1.3.2.2 Δευτερογενή ή συναδροισμένα. Είναι τα δεδομένα που παράγονται από τα πρωτογενή ή από άλλα δευτερογενή όταν αυτά συναδροιστούν { Γ } σε δεδομένα μικρότερης χρονικής διακριτότητας, όταν αλλάζουν δηλαδή χρονική διακριτότητα, πχ. όταν ημερήσια δεδομένα συναδροιστούν σε μηνιαία. Η συνάδρωση αυτή μπορεί να γίνει με τρεις τρόπους, ανάλογα με την αντίστοιχη μεταβλητή και τα χαρακτηριστικά του οργάνου:

- με άθροιση των τιμών μεγαλύτερης διακριτότητας για διάστημα αντίστοιχο με αυτό της μικρότερης διακριτότητας, όπως γίνεται πχ. για τη βροχή.

- με εξαγωγή κάποιας μέσης τιμής για το αντίστοιχο διάστημα, όπως πχ. για τη θερμοκρασία (βλ. παρακάτω για τον τρόπο υπολογισμού της μέσης τιμής).
- με απλή απομόνωση κάποιας συγκεκριμένης τιμής (συνήθως αρχικής ή τελικής), δηλαδή με “στιγμιαίο” τρόπο, όπως πχ. για την παροχή.

Ενώ τα πρωτογενή δεδομένα “μετρούνται” και αποθηκεύονται στη ΒΔ σε όποια χρονική διακριτότητα “μετρήθηκαν”, τα δευτερογενή δημιουργούνται (και μπορούν να αποθηκεύονται στη ΒΔ) σε σταθερές - εκ των προτέρων γνωστές - πρότυπες { Γ } διακριτότητες: 1 - 5 - 10 - 30 λεπτά, 1 - 3 - 6 - 8 - 12 ώρες, 1 - 15 ημέρες, 1 - 6 μήνες, έτος. Αυτή τη στιγμή μάλιστα αποθηκεύονται στη ΒΔ τα δευτερογενή δεδομένα μόνο για τις διακριτότητες: ώρα, ημέρα, μήνας, έτος. Για τις άλλες απλά μπορούν να δημιουργηθούν, όχι όμως και να αποθηκευθούν, αν υπάρχουν τα κατάλληλης διακριτότητας πρωτογενή, πχ. να δημιουργηθούν 10λεπτα δευτερογενή αν υπάρχουν πρωτογενή 5λεπτα ή ενός λεπτού.



Σχήμα 2.1.3: Σχέση επιπέδων και πρωτογενών - δευτερογενών χρονοσειρών

Τα δευτερογενή δεδομένα κατατάσσονται επίσης σε 3 επίπεδα, ανάλογα με τα δεδομένα από τα οποία προέρχονται. Δευτερογενή ενός επιπέδου μπορούν να προέρχονται είτε από άλλα δεδομένα (πρωτογενή ή άλλα δευτερογενή) μεγαλύτερης χρονικής διακριτότητας είτε (σπανιότερα) από δευτερογενή ίδιας

διακριτότητας αλλά του προηγούμενου επιπέδου, στα οποία έχουν εφαρμοσθεί οι απαραίτητες διαδικασίες ελέγχου - μεμονωμένης συμπλήρωσης - χαρακτηρισμού (προκειμένου για επίπεδο 2 από 1) ή εκτενούς συμπλήρωσης - ομογενοποίησης (προκειμένου για επίπεδο 3 από 2). Έτσι για παράδειγμα, μηνιαία δευτερογενή επιπέδου 2 μπορούν να προέρχονται από (πρωτογενή ή άλλα δευτερογενή) ημερήσια δεδομένα επιπέδου 2 ή από μηνιαία δευτερογενή επιπέδου 1.

2.2 Μορφές Αποθήκευσης Χρονοσειρών

Το κυριότερο χαρακτηριστικό των διαδικασιών συλλογής και επεξεργασίας υδρολογικών, υδρογεωλογικών και μετεωρολογικών δεδομένων, τόσο σε σχέση με τους Φορείς που συμμετέχουν στο ΥΔΡΟΣΚΟΠΙΟ όσο και γενικότερα στον Ελλαδικό χώρο, είναι πως δεν υπάρχει κανενός είδους προτυποποίηση { Γ } αυτών των διαδικασιών. Το δείγμα των μετρούμενων μεταβλητών, η συχνότητα μέτρησης, οι τρόποι υπολογισμού διαφέρουν από Φορέα σε Φορέα, μεταξύ διαφορετικών ομάδων εργασίας του ίδιου Φορέα ή ακόμα και ανάλογα με το είδος ή την ταυτότητα του μετρητικού σταθμού. Κατά συνέπεια, οποιαδήποτε προσπάθεια απεικόνισης σε μια ΒΔ αυτής της κατάστασης με τρόπους που προϋποθέτουν οποιοδήποτε είδους προτυποποίηση είναι εκ των πραγμάτων δυσχερής. Για παράδειγμα, έστω πως όλοι οι μετεωρολογικοί σταθμοί μετρούν τα μεγέθη θερμοκρασία, άνεμο, πίεση, υγρασία, οπότε θα ήταν δυνατό να αποθηκεύονται για κάθε σταθμό και για κάθε μέτρηση μόνο τα 4 αυτά μεγέθη. Όμως κάτι τέτοιο θα οδηγούσε αναμφίβολα είτε σε απώλεια πληροφορίας, στην περίπτωση που κάποιοι μετεωρολογικοί σταθμοί μετρούν κάποια επιπλέον μεγέθη, είτε σε σπατάλη χώρου, αν μετρούνται ορισμένα μόνο από αυτά τα μεγέθη. Μια δυσκολότερη περίπτωση είναι η υπόθεση πως κάποιος συγκεκριμένος σταθμός μετρά ένα μέγεθος με μια ορισμένη συχνότητα. Στην περίπτωση αυτή θα αρκούσε να αποθηκευθεί μια ημερομηνία έναρξης των μετρήσεων και στη συνέχεια μια σειρά αριθμών που θα ήταν οι πραγματικές μετρήσεις. Ωστόσο κάτι τέτοιο δεν είναι δυνατό εφόσον ο σταθμός διακόπτει σε τυχαία χρονικά διαστήματα τη λειτουργία του ή αλλάζει η συχνότητα των παρατηρήσεων.

Για να ξεπεραστούν αυτά τα προβλήματα ορίζεται πως οποιαδήποτε εγγραφή χρονοσειράς ταυτοποιείται καταρχήν από το όργανο και την ημερομηνία στα οποία αναφέρεται. Τα πρωτογενή δεδομένα έχουν εξ ορισμού { Γ } ένα όργανο και μια ημερομηνία αναφοράς: είναι το (πραγματικό ή παράγωγο) όργανο που "μέτρησε" τη συγκεκριμένη τιμή και η ημερομηνία⁶ κατά την οποία έγινε η μέτρηση. Τα δευτερογενή δεδομένα προέρχονται από αντίστοιχα πρωτογενή και κατά συνέπεια αναφέρονται στο όργανο των πρωτογενών. Όσο

⁶ Στο εξής με τη λέξη "ημερομηνία" εννοείται "έτος - μήνας - ημέρα - ώρα - λεπτά".

για την ημερομηνία, αυτή δεν είναι πραγματική ημερομηνία μέτρησης, είναι ωστόσο μια συμβατική ημερομηνία που αντιστοιχεί στο χρονικό διάστημα για το οποίο έγινε η συνάδρωση. Το ποιά είναι αυτή η ημερομηνία εξαρτάται από τη χρονική κλίμακα και το είδος της μεταβλητής.

Επίσης κάθε εγγραφή πρέπει να συνοδεύεται από κάποια στοιχεία σχετικά με την "ιστορία" της, όπως το ποιοί έλεγχοι έχουν πραγματοποιηθεί πάνω στην εγγραφή και ποιά ήταν το αποτέλεσμα του καθενός [Ανα92], καθώς και γενικότερα χαρακτηριστικά της, όπως το επίπεδο στο οποίο ανήκει. Τα στοιχεία αυτά επίσης πρέπει να περιληφθούν σε όσα ταυτοποιούν την εγγραφή. Πράγματι, για ένα συγκεκριμένο συνδυασμό όργανου και ημερομηνίας μπορεί να υπάρχουν δύο ή και περισσότερες εγγραφές: μία αυτή της πρωτογενούς μέτρησης και μία αυτή της διορθωμένης τιμής, σε περίπτωση που διαπιστωθεί πως η μέτρηση είχε κάποιο λάθος. Συνεπώς η μορφή αποθήκευσης των χρονοσειρών γίνεται:

<όργανο><ημερομηνία><χαρακτηριστικά><τιμή>

Η τιμή με την σειρά της μπορεί να αποτελείται από περισσότερα από ένα πεδία. Αυτό συμβαίνει διότι πολλά όργανα μετρούν ταυτόχρονα με μία μέτρηση δύο ή περισσότερα φυσικά μεγέθη. Πχ. ένα όργανο μέτρησης ανέμου μετρά τόσο τη διεύθυνση όσο και την ταχύτητα του ανέμου. Αρα τελικά στην παραπάνω παράσταση θα είναι⁷:

<τιμή> = <πεδίο 1> [<πεδίο-2>...<πεδίο-κ>]

και πρέπει να καθορισθεί ποιά θα είναι η τιμή του κ για κάθε τύπο οργάνου και επίσης ποιά θα είναι το εύρος κάθε πεδίου.

2.3 Κατανεμημένη λειτουργία

Ο σχεδιασμός και η λειτουργία της ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ περιπλέκεται από το γεγονός πως είναι κατανεμημένη και κατά συνέπεια υπόκειται σε ορισμένους περιορισμούς ως προς τη διακίνηση και το χειρισμό των δεδομένων. Υπάρχουν επίσης ιδιαίτερες απαιτήσεις που απορρέουν από τη φύση και τη μορφή του Έργου. Οι περιορισμοί και οι απαιτήσεις έχουν ως εξής:

2.3.1 Μειωμένη ταχύτητα. Το δίκτυο του ΥΔΡΟΣΚΟΠΙΟΥ έχει σχεδιασθεί με στόχο την ταχύτερη δυνατή λειτουργία [Παπ92]. Ωστόσο, οι ταχύτητες που επιτυγχάνονται είναι της τάξης των Kbits/sec, που είναι πολύ μικρότερες από τις ταχύτητες μεταφοράς δεδομένων από τοπικούς δίσκους στην κεντρική μνήμη που μετρώνται σε Mbits/sec. Κατά

⁷ "[...]" σημαίνει πως ό,τι (...) περιέχεται μέσα στις αγκύλες είναι προαιρετικό

συνέπεια, οι εφαρμογές πρέπει να λαμβάνουν υπόψη τους πως, καθώς δεν απευθύνονται σε τοπική αλλά σε κατανεμημένη ΒΔ, δεν έχουν στη διάθεση τους απεριόριστη ταχύτητα ανάκτησης δεδομένων. Η πρόσβαση στα δεδομένα πρέπει να είναι:

2.3.1.1 Καθορισμένη και συγκροτημένη. Δηλαδή να γίνεται μόνο όταν πραγματικά χρειάζεται. Επίσης, να ομαδοποιούνται διαδοχικές αιτήσεις ανάκτησης σε μία.

2.3.1.2 Περιορισμένη. Δηλαδή να ανακτώνται μόνο τα δεδομένα που πραγματικά χρειάζονται. Για παράδειγμα η τακτική του να ανακτώνται αρχικά "όλα" τα δεδομένα και να απορρίπτονται στη συνέχεια όσα δεν χρειάζονται είναι ιδιαίτερα επιβαρυντική στα πλαίσια της κατανεμημένης ΒΔ. Για αυτό το λόγο θα πρέπει να χρησιμοποιούνται κατάλληλα οι διαθέσιμες διαχειριστικές πληροφορίες για εντοπισμό εκ των προτέρων του συνόλου των πληροφοριών που ζητούνται.

2.3.1.3 Οικονομική. Ανάμεσα σε διαφορετικούς ισοδύναμους τύπους δεδομένων να ανακτώνται οι μικρότεροι σε μέγεθος και να μετατρέπονται, αν υπάρχει ανάγκη, από την εφαρμογή. Για παράδειγμα, αν ένας τύπος δεδομένων μπορεί να μεταφερθεί ως ακέραιος 4 Bytes αλλά έχει και μια εναλλακτική δυνατότητα για μεταφορά ως συμβολοσειρά { Γ } 25 χαρακτήρων, τότε θα πρέπει προφανώς να προτιμηθεί η μεταφορά ως ακέραιου. Αυτή η διαδικασία έχει το μειονέκτημα πως, σε περίπτωση που τελικά χρειάζεται η συμβολοσειρά, θα πρέπει να γίνει η μετατροπή από την αντίστοιχη εφαρμογή. Αλλά μειώνει κατά 21 Bytes τον αριθμό των μεταφερόμενων δεδομένων, για κάθε μεταφερόμενη εγγραφή.

2.3.2 Απλές αιτήσεις. Οι αιτήσεις προς τη ΒΔ είναι γενικά απλές, δεν περιλαμβάνουν δηλαδή πολύπλοκες εκφράσεις που απαιτούν το συνδυασμό και την ένωση { Γ } πληροφοριών από πολλά διαφορετικά αντικείμενα. Συνήθως έχουν τη μορφή της ανάκτησης πληροφοριών με κάποια χαρακτηριστικά για κάποιο όργανο και μια συγκεκριμένη χρονική περίοδο. Για τέτοιες απλές αιτήσεις το μεγαλύτερο μέρος του απαιτούμενου χρόνου εξυπηρέτησης κάποιας αίτησης είναι ο χρόνος που χρειάζονται τα δεδομένα της απάντησης να μεταφερθούν πάνω από το δίκτυο, μια και ο χρόνος επεξεργασίας της αίτησης στην ΤΒΔ είναι αρκετά μικρός. Αντίστοιχα μειώνεται η σημασία παραδοσιακών τεχνικών αύξησης της ταχύτητας όπως η βελτιστοποίηση των αιτήσεων { Γ } και η σωστή επιλογή δεικτών { Γ }, χωρίς αυτό βέβαια να σημαίνει πως δεν δόθηκε σε αυτά τα σημεία η δέουσα προσοχή.

2.3.3 Διαχειριστική αυτονομία. Κάθε Φορέας διατηρεί την αυτονομία του και διαχειρίζεται ο ίδιος τα δεδομένα του. Κατά συνέπεια, μόνο ο Φορέας είναι αποκλειστικά

υπεύθυνος για την εισαγωγή και την ενημέρωση τόσο των διαχειριστικών πληροφοριών όσο και των δεδομένων (στατικών και χρονοσειρών) που του "ανήκουν". Αρα δεν είναι δυνατή λειτουργία ενημέρωσης πάνω από την ΚΒΔ. Σε κάθε περίπτωση που απαιτείται ενημέρωση αυτή πρέπει να γίνεται τοπικά. Ως εκ τούτου, πάνω από το δίκτυο και στην ΚΒΔ το ποσοστό των αιτήσεων ανάγνωσης είναι 100% του συνόλου των αιτήσεων. Εκτός αυτού, η απάντηση για αιτήσεις χρονοσειρών και σταθερών στην ΚΒΔ βρίσκεται, αν υπάρχει, πάντα σε έναν και μόνο κόμβο, αυτόν του Φορέα - διαχειριστή τους. Επιπλέον, αιτήσεις χρονοσειρών και σταθερών στην ΤΒΔ ικανοποιούνται από το σύνολο των πληροφοριών του τοπικού κόμβου και άρα του τοπικού Φορέα - διαχειριστή, ενώ ίδιες αιτήσεις προς την ΚΒΔ ικανοποιούνται από το σύνολο των πληροφοριών του ΥΔΡΟΣΚΟΠΙΟΥ. Ο χρήστης θα μπορεί, μέσα από τις εφαρμογές, να καθορίσει ποιό είδος πρόσβασης επιθυμεί.

2.3.4 Αντιγραφή διαχειριστικών πληροφοριών. Για να διευκολυνθεί η ορθή ανάκτηση των δεδομένων είναι απαραίτητο να προηγηθεί επεξεργασία σημαντικού όγκου διαχειριστικών πληροφοριών, ώστε να εντοπιστούν τα ζητούμενα διαδέσιμα δεδομένα. Αυτή η διαδικασία θα ήταν υπερβολικά αργή στην ΚΒΔ. Για να έρδει η ταχύτητα του συστήματος σε ρεαλιστικά επίπεδα επιλέγεται η λύση της αντιγραφής { Γ } των διαχειριστικών πληροφοριών. Κάθε Φορέας είναι βέβαια αποκλειστικά υπεύθυνος για την εισαγωγή και διαχείριση των δικών του διαχειριστικών πληροφοριών. Ωστόσο, κάθε βράδυ, σε ώρες δηλαδή εκτός αιχμής, οι διαχειριστικές πληροφορίες που εισήχθησαν ή ενημερώθηκαν κατά τη διάρκεια της ημέρας σε κάθε κόμβο αντιγράφονται σε όλους τους άλλους κόμβους του δικτύου, έτσι ώστε όλοι οι κόμβοι να διατηρούν το σύνολο των διαχειριστικών πληροφοριών της ΚΒΔ. Κατά συνέπεια, κάθε πρόσβαση διαχειριστικών πληροφοριών σε οποιοδήποτε κόμβο μετατρέπεται σε τοπική πρόσβαση. Κάτι τέτοιο έχει μικρό κόστος σε χώρο στο δίσκο, μια και οι διαχειριστικές πληροφορίες είναι λίγες σε σχέση με τα δεδομένα. Έχει βέβαια και κάποιο κόστος σε επιπλέον κυκλοφορία πληροφοριών στο δίκτυο. Αυτή όμως η κυκλοφορία "μετατίθεται" από τις ώρες αιχμής, οπότε επηρεάζει και καθυστερεί τη λειτουργία των εφαρμογών και γενικότερα τη χρήση του συστήματος σε ώρες εκτός αιχμής. Το αποτέλεσμα είναι μεγάλη τελικά αύξηση της ταχύτητας χρήσης του συστήματος. Η χρήση του συστήματος δεν θα ήταν πρακτικά δυνατή -από άποψη ταχύτητας απόκρισης- σε περίπτωση μη χρησιμοποίησης αυτής της μεθοδολογίας.

Αυτό που γίνεται τελικά στις εφαρμογές είναι το εξής: κάθε εφαρμογή διατηρεί δύο παράλληλες συνδέσεις { Γ }, μια με την ΤΒΔ και μια με την ΚΒΔ. Κάθε πρόσβαση σε διαχειριστικές πληροφορίες ή πληροφορίες εφαρμογών γίνεται από τη σύνδεση με την ΤΒΔ. Μόνο τη στιγμή που πρόκειται να γίνει πρόσβαση σε δεδομένα (χρονοσειρές ή σταθερές) αλλάζει η ενεργή σύνδεση σε αυτή της ΚΒΔ. Γίνεται η ερώτηση στην ΚΒΔ και

στη συνέχεια, αφού επιστραφεί απάντηση, επιστρέφει η ενεργή σύνδεση στην ΤΒΔ. Με αυτό τον τρόπο, για την πλειονότητα των περιπτώσεων οι προσβάσεις στη ΒΔ είναι τοπικές, εκτός της πρόσβασης στα δεδομένα. Τα τελευταία βέβαια δεν αντιγράφονται σε κάθε κόμβο.

3 ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΥΛΟΠΟΙΗΣΗΣ

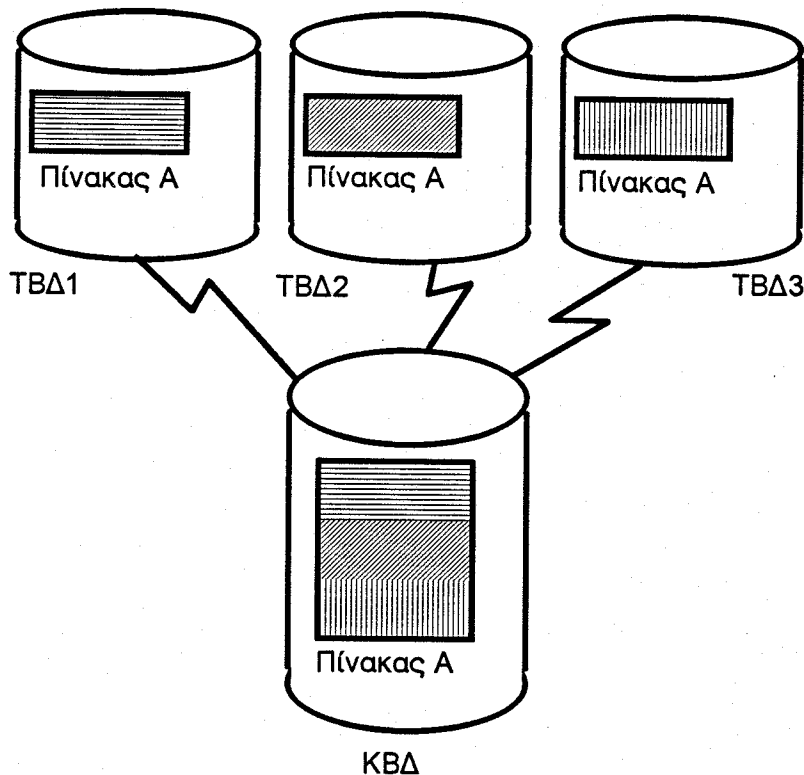
3.1 Τοπική βάση δεδομένων

Δημιουργείται σε κάθε κόμβο, ως ΤΒΔ, μια σχεσιακή { Γ } ΒΔ με όνομα db0. Όλες οι ΤΒΔ έχουν το ίδιο σχήμα { Γ }. Στους πίνακες { Γ } της αποθηκεύονται οι πληροφορίες όλων των κατηγοριών. Η γλώσσα SQL { Γ } [SQL86][INGRES91a] είναι η γλώσσα ορισμού και χειρισμού των δεδομένων { Γ }. Όψεις { Γ } πάνω στους πίνακες προσφέρουν μια εναλλακτική θεώρηση των δεδομένων. Οι όψεις αυτές στην πλειοψηφία τους δεν δημιουργούνται από πολύπλοκες εκφράσεις SQL αλλά από απλές πράξεις λογικής ένωσης { Γ } όπου κάθε ένας από τους συμμετέχοντες πίνακες είναι μια οριζόντια διαμέριση { Γ } της όψης. Για παράδειγμα, έστω τρεις πίνακες A, B και C με κοινή μορφή και έστω μια όψη D που αποτελεί λογική ένωσή τους. Αυτό σημαίνει πως η όψη D "περιέχει" -εντός εισαγωγικών διότι η όψη δεν περιέχει τίποτα από μόνη της, μια και είναι ένα εικονικό κατασκεύασμα- όλα τα στοιχεία των A, B και C και έχει επίσης κοινή μορφή με αυτούς. Το αν ένα αντικείμενο στη ΒΔ είναι όψη ή πίνακας είναι διαφανές στο χρήστη και τον προγραμματιστή όταν πρόκειται για ανάκτηση πληροφοριών. Ωστόσο, στη γενική περίπτωση, πάνω στις όψεις δεν επιτρέπεται ενημέρωση - εισαγωγή - διαγραφή πληροφοριών. Αυτές οι πράξεις πρέπει να γίνουν απευθείας στους υποκείμενους { Γ } πίνακες.

3.2 Κατανεμημένη βάση δεδομένων

Η ΚΒΔ ονομάζεται ddb0 και προσπελαύνεται ως ddb0/d, πχ. sql ddb0/d, για να διακρίνεται ακριβώς το γεγονός πως είναι κατανεμημένη. Νοηματικά η ΚΒΔ αποτελεί τη λογική ένωση των διαφόρων συμμετεχουσών ΤΒΔ. Παρ' ό,τι η ΚΒΔ είναι και αυτή σχεσιακή ως προς την αρχιτεκτονική της, δεν περιέχει πίνακες. Στην πραγματικότητα η ΚΒΔ είναι μια εικονική οντότητα: τα αντικείμενα της δεν είναι πραγματικά αλλά σχετίζονται με κάποιο τρόπο με τα αντίστοιχα αντικείμενα των ΤΒΔ: είναι (τουλάχιστον στην υλοποίηση του ΥΔΡΟΣΚΟΠΙΟΥ) είτε όψεις είτε σύνδεσμοι { Γ } [INGRES91b]. Ένα αντικείμενο A στην ΚΒΔ ddb0 είναι όψη όταν είναι ένα εικονικό κατασκεύασμα που αποτελεί τη λογική ένωση όλων των αντικειμένων A (όψεων ή πινάκων) όλων των συμμετεχουσών ΤΒΔ. Κατά αυτό τον τρόπο το αντικείμενο A περιέχει το σύνολο των πληροφοριών όλων των τοπικών αντικειμένων A των ΤΒΔ, δηλαδή τελικά το σύνολο των πληροφοριών του ΥΔΡΟΣΚΟΠΙΟΥ που περιέχονται στα A. Και μια ερώτηση στο A της ΚΒΔ αυτόματα ανακτά τις πληροφορίες που την ικανοποιούν από όλες τις συμμετέχουσες ΤΒΔ. Ένα αντικείμενο B στην ΚΒΔ ddb0 είναι σύνδεση όταν είναι απευθείας "εικόνα" του αντίστοιχου αντικειμένου B μιας ΤΒΔ. Έτσι, ενώ οι όψεις της ΚΒΔ αναπαριστούν ως επί το

πλείστον δυναμικές πληροφορίες (διαχειριστικές και δεδομένα) από όλους τους κόμβους, οι συνδέσεις αναπαριστούν κυρίως στατικές πληροφορίες (εφαρμογών - βοηθητικές) που είναι ίδιες σε όλους τους κόμβους.



Σχήμα 3.2.1: Γενική μορφή αντικειμένων KBΔ

Με κατάλληλη χρήση των συνδέσεων και των όψεων η KBΔ έχει ακριβώς το ίδιο σχήμα με όλες τις TBΔ: τα ίδια αντικείμενα με την ίδια μορφή. Αρα οι εφαρμογές δε χρειάζεται να κάνουν διάκριση: μια ερώτηση που λειτουργεί στην TBΔ θα λειτουργεί απaráλλαχτη και στην KBΔ -παρ' ό,τι δεν θα δίνει αναγκαστικά τα ίδια αποτελέσματα (αφού οι τοπικές πληροφορίες είναι γνήσιο υποσύνολο του συνόλου των πληροφοριών). Πχ. μια ερώτηση σε ένα τοπικό πίνακα A θα δώσει ως απάντηση μόνο τις τοπικές πληροφορίες, ενώ σε μια όψη A της KBΔ θα δώσει το σύνολο των πληροφοριών του ΥΔΡΟΣΚΟΠΙΟΥ που ικανοποιούν την ερώτηση. Έτσι επιτυγχάνεται και η διαφάνεια τοποθέτησης $\{ \Gamma \}$: οποιαδήποτε ερώτηση προς κάποιο αντικείμενο της KBΔ αυτόματα μετατρέπεται σε ίδιες υποερωτήσεις προς τα διάφορα αντικείμενα των TBΔ από τα οποία απαρτίζεται το αντικείμενο της KBΔ. Ούτε ο προγραμματιστής εφαρμογών ούτε, πολύ περισσότερο, ο χρήστης χρειάζεται να γνωρίζουν σε ποιά ακριβώς TBΔ είναι αποθηκευμένη η ζητούμενη πληροφορία. Σημασία εδώ έχει το γεγονός πως τα δεδομένα του ΥΔΡΟΣΚΟΠΙΟΥ αποθηκεύονται και προσπελαύνονται από

την ΚΒΔ με τέτοιο τρόπο ώστε κάθε πληροφορία να βρίσκεται πάντα σε έναν και μόνο κόμβο. Πχ. οι πληροφορίες για ένα σταθμό βρίσκονται (στην ΚΒΔ) πάντα στο Φορέα που τον διαχειρίζεται και αντίστοιχα και τα δεδομένα του σταθμού αυτού.

3.3 Μορφή ΚΒΔ

Από τους κόμβους του δικτύου του ΥΔΡΟΣΚΟΠΙΟΥ ορισμένοι μόνο έχουν δεδομένα: ΕΜΥ, ΕΜΠ, ΥΠΓΕ, ΥΠΕΧΩΔΕ, ΕΑΑ, ΔΕΗ. Η ΚΒΔ σχηματίζεται έτσι ώστε να συμμετέχουν σε αυτή μόνο οι ΤΒΔ αυτών των κόμβων. Με αυτό τον τρόπο σε μια ερώτηση προς την ΚΒΔ είναι σίγουρο πως δεν θα αναζητηθούν οι πληροφορίες σε σημείο όπου σίγουρα δεν υπάρχουν (σε κάποιους κόμβους χωρίς δεδομένα). Η συμμετοχή στην ΚΒΔ αυξάνει το φόρτο κάθε ΤΒΔ -τόσο του εξυπηρετητή { Γ } όσο και γενικότερα του συστήματος-. Αντίστοιχα, τυχόν συμμετοχή ΤΒΔ χωρίς δεδομένα θα αύξανε χωρίς λόγο το χρόνο επεξεργασίας αιτήσεων, μια και οι αιτήσεις αυτές θα μεταβιβάζονταν και σε αυτή την ΤΒΔ, για να πάρουν φυσικά στο τέλος "αρνητική" απάντηση.

Η ΚΒΔ έχει έναν κόμβο - συντονιστή { Γ } που αποθηκεύει την περιγραφή των αντικειμένων της -τα ίδια τα αντικείμενα είναι εικονικά-. Στον κόμβο - συντονιστή "τρέχει" ο εξυπηρετητής της ΚΒΔ (iistar) και τέλος στον κόμβο αυτό συγκεντρώνονται οι "υποαπαντήσεις" από τις διάφορες ΤΒΔ για τελική επεξεργασία πριν την αποστολή τους ως απάντηση σε κάποια ερώτηση. Λόγω του τελευταίου χαρακτηριστικού, σε περίπτωση που υπάρχει ένας μόνο συντονιστής υπάρχει περίπτωση άσκοπης κυκλοφορίας στην περίπτωση που υπάρχει πρόσβαση σε δεδομένα της ΤΒΔ αλλά μέσα από την ΚΒΔ. Για παράδειγμα, έστω ότι ο συντονιστής βρίσκεται στην ΕΜΥ. Μια ερώτηση στην ΚΒΔ από πχ. το ΥΠΕΧΩΔΕ που αφορά δεδομένα του ΥΠΕΧΩΔΕ θα απαντηθεί ως εξής: η ερώτηση θα υποβληθεί στο συντονιστή στην ΕΜΥ, αυτός θα στείλει την ερώτηση σε όλους τους συμμετέχοντες κόμβους, θα πάρει απάντηση από το ΥΠΕΧΩΔΕ, δεδομένα θα κινηθούν από το ΥΠΕΧΩΔΕ προς την ΕΜΥ, όπου θα γίνει κάποια τελική επεξεργασία και θα επιστραφούν στο ΥΠΕΧΩΔΕ. Προφανώς η κίνηση ΥΠΕΧΩΔΕ - ΕΜΥ - ΥΠΕΧΩΔΕ είναι άσκοπη. Το ενδεχόμενο αυτό είναι αρκετά πιθανό, αφού σε κάθε κόμβο θα γίνεται σε μεγάλο βαθμό επεξεργασία τοπικών δεδομένων. Για να μειωθεί η επίδραση παρόμοιων περιπτώσεων στη λειτουργία του δικτύου θα υπάρχουν 6 κόμβοι - συντονιστές στους κόμβους δεδομένων: ΕΜΠ, ΕΜΥ, ΥΠΓΕ, ΥΠΕΧΩΔΕ, ΕΑΑ, ΔΕΗ. Κάθε φορά θα γίνεται σύνδεση κάποιας εφαρμογής στην ΚΒΔ στον κοντινότερο κόμβο - συντονιστή, δηλαδή, προκειμένου για κόμβους δεδομένων στον ίδιο τον κόμβο και προκειμένου για τους άλλους κόμβους στον κόμβο του ΕΜΠ, λόγω της μορφής του δικτύου του ΥΔΡΟΣΚΟΠΙΟΥ. Ετσι υπάρχουν "φυσικά" 6 ΚΒΔ, ωστόσο "λογικά" υπάρχει μια μόνο ΚΒΔ, αφού όλες αυτές οι ΚΒΔ έχουν ίδιο σχήμα, προσπελαύνουν όλες και τις 6 ΤΒΔ με δεδομένα με τον ίδιο τρόπο και διαφέρουν μόνο στην τοποθέτηση του κόμβου - συντονιστή. Για το λόγο αυτό στη

συνέχεια σε κάθε αναφορά στην ΚΒΔ εννοείται η μία "λογική" ΚΒΔ, άσχετα από τον χρησιμοποιούμενο κόμβο - συντονιστή.

3.4 Ταυτοποίηση αντικειμένων

Η ταυτοποίηση { Γ } των διάφορων αντικειμένων της ΒΔ γίνεται με βάση τον Κωδικό ΥΔΡΟΣΚΟΠΙΟΥ". Αυτός είναι ένας μοναδικός κωδικός για τα βασικά αντικείμενα της ΒΔ, τους σταθμούς και τα όργανα. Τα δεδομένα αναφέρονται πάντα σε κάποιο όργανο το οποίο, με τη σειρά του, ανήκει σε ένα και μοναδικό σταθμό. Αρα, ο Κωδικός Σταθμού είναι η αρχική ταυτοποίηση των αντικειμένων.

3.4.1 Ταυτοποίηση σταθμών - οργάνων. Κάθε σταθμός, ανάλογα με τον Φορέα στον οποίο ανήκει έχει ένα αύξοντα αριθμό. Κάθε Φορέας αντιστοιχίζεται σε μια περιοχή αριθμών -το ΕΑΑ στην πρώτη περιοχή για ιστορικούς λόγους και το ΥΠΓΕ στην τελευταία λόγω μεγάλου και σε μεγάλο βαθμό ακαθόριστου πλήθους σταθμών. Σε κάθε νέο σταθμό διατίθεται ο επόμενος διαθέσιμος αύξοντας αριθμός στη περιοχή. Ο α/α δεν έχει καμία περαιτέρω σημασία, είναι απλά συνάρτηση της σειράς εισαγωγής του σταθμού στην ΤΒΔ. Η εισαγωγή νέου σταθμού και η ανάθεση νέου αριθμού γίνεται από τη ΔΒΔ { Γ } [INGRES91a][INGRES91d] _station() και μόνο από αυτήν. Δεν επιτρέπεται απευθείας, χωρίς χρήση των αντίστοιχων ΔΒΔ, εισαγωγή στους πίνακες σταθμών, οργάνων, χρονοσειρών, σταθερών και γεγονότων. Οι ΔΒΔ είναι αντίστοιχα οι _station(), _instrument(), _timeseries(), _constant() και _event().

Ο τελικός κωδικός ΥΔΡΟΣΚΟΠΙΟΥ του σταθμού είναι, σύμφωνα και με τον πίνακα 3.4.1.1:

Κωδικός ΥΔΡΟΣΚΟΠΙΟΥ = (αρχή περιοχής αύξοντων αριθμών +
αύξων αριθμός του σταθμού στο Φορέα) * 256

πχ. ο 4ος σταθμός του ΥΠΕΧΩΔΕ έχει κωδικό ΥΔΡΟΣΚΟΠΙΟΥ $(30000 + 4) * 256 = 7681024$. Στον πίνακα φαίνεται πως η περιοχή αύξοντων αριθμών που αντιστοιχίζεται σε κάθε Φορέα είναι άμεση συνάρτηση του πλήθους των σταθμών που διαδέχεται και, σε κάθε περίπτωση, πολύ μεγαλύτερη από το πραγματικό πλήθος των σταθμών του Φορέα. Ορισμένοι Φορείς έχουν μικρό αριθμό σταθμών, άλλοι ένα μόνο σταθμό και άλλοι κανέναν -αλλά ίσως αποκτήσουν στο μέλλον. Υπάρχουν επίσης αδιάθετες περιοχές, για μελλοντικούς συμμετέχοντες Φορείς.

Δοθέντος του κωδικού σταθμού, είναι απλό να αντιστοιχίζονται και μοναδικοί κωδικοί οργάνων. Αφού κάθε όργανο ανήκει σε έναν σταθμό, τότε ο α/α του οργάνου στο σταθμό (ο α/α έχει να κάνει -όπως και με το σταθμό- με την εισαγωγή των στοιχείων του οργάνου στην ΤΒΔ του αντίστοιχου Φορέα) προστιδόμενος στον κωδικό του σταθμού δίνει τον

κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του οργάνου. Κατά συνέπεια, αν ένας κωδικός αντικειμένου διαιρείται με το 256 τότε είναι σταθμός, αλλιώς είναι όργανο. Τα όργανα ενός σταθμού με κωδικό Σ μπορεί να είναι μέχρι 127 τον αριθμό, με κωδικούς από Σ + 1 ως Σ + 127.

Φορέας	Περιοχή αύζοντων αριθμών	Περιοχή τελικών κωδικών ΥΔΡΟΣΚΟΠΙΟΥ
ΕΑΑ	0 - 9999 [*]	0 - 2559744
ΕΜΥ	10000 - 29999	2560000 - 7679744
ΥΠΕΧΩΔΕ	30000 - 49999	7680000 - 12799744
ΔΕΗ	50000 - 69999	12800000 - 17919744
ΕΜΠ	70000 - 70999	17920000 - 18175744
ΑΠΘ/ΤΥΤΠ	71000 - 71999	18176000 - 18431744
ΑΠΘ/ΕΤ	72000 - 72999	18432000 - 18687744
ΕΚΠΑ	73000 - 73999	18688000 - 18943744
ΕΚΕΦΕ"Δ"	74000 - 74999	18944000 - 19199744
ΥΒΕΤ	80000 - 84999	20480000 - 21759744
ΕΥΔΑΠ	90000 - 94999	23040000 - 24319744
ΥΠΓΕ	100000 -	25600000 -

Πίνακας 3.4.1.1: Οι συμμετέχοντες Φορείς και οι διαδέσιμες περιοχές αύζοντων αριθμών και κωδικών ΥΔΡΟΣΚΟΠΙΟΥ. Οι κωδικό ΥΔΡΟΣΚΟΠΙΟΥ προκύπτουν με πολλαπλασιασμό του αύζοντος αριθμού με το 256.

Τελικά οι κωδικό ΥΔΡΟΣΚΟΠΙΟΥ επιτρέπουν ως 2147483647 σταθμούς και όργανα στη ΒΔ. Υλοποιούνται σαν ένας ξεχωριστός τύπος δεδομένων οριζόμενος από το χρήστη { Γ } [INGRES91i], hsiid (HYDROSCOPE station & instrument identifier). Στην πραγματικότητα είναι integer4 (ακέραιος 4 Bytes) και "συμπεριφέρονται" σαν integer4. Με αυτή την υλοποίηση κάθε σταθμός και όργανο στη ΒΔ έχει μια μοναδική ταυτότητα, η οποία μάλιστα συνδέεται και με το Φορέα που το διαχειρίζεται. Ταυτόχρονα γίνεται και οικονομία στο χώρο των δίσκων και σε κυκλοφορία στο δίκτυο. Σημαντικό στοιχείο για τις εφαρμογές είναι η απαίτηση αυτός ο κωδικός να μην μπορεί, σε καμία περίπτωση, να τροποποιηθεί από το χρήστη, μια και ο χειρισμός του γίνεται αυτόματα από το σύστημα. Αλλά και επειδή ο

κωδικός είναι ένας μεγάλος ακέραιος (1 ως 8 ψηφία) ο οποίος μπορεί να μπερδέγει κάποιο άπειρο χρήστη, προτείνεται να μην εμφανίζεται από τις εφαρμογές στις οδόνες των χρηστών, εκτός και αν αυτοί επιλέξουν να τον βλέπουν.

3.4.2 Ταυτοποίηση χρονοσειρών. Τα δεδομένα χρονοσειρών αποθηκεύονται με βάση τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του πραγματικού όργανου στο οποίο αναφέρονται με βάση τη βασική μορφή:

<όργανο><ημερομηνία><χαρακτηριστικά><τιμή>

Μόνο όμως ο κωδικός του οργάνου δεν αρκεί για να ταυτοποιήσει μοναδικά τη συγκεκριμένη εγγραφή. Χρειάζεται να περιληφθεί και η ημερομηνία στο σύνολο των πεδίων ταυτοποίησης, μια και κάθε εγγραφή αναφέρεται σε μια "μέτρηση" του οργάνου για κάποια συγκεκριμένη ημερομηνία.

Υπάρχει η περίπτωση μια τιμή σε κάποιο επίπεδο για κάποιο λόγο να μην περιλαμβάνεται στο επόμενο επίπεδο, οπότε τελικά για το ίδιο ζεύγος κωδικού οργάνου - ημερομηνίας υπάρχουν περισσότερες από μία εγγραφές, με διαφορετικές τιμές. Για παράδειγμα, η τιμή είναι επιπέδου 1 αλλά λανθασμένη οπότε στο επίπεδο 2 περιλαμβάνεται η σωστή τιμή με ίδιο κωδικό και ημερομηνία, ή είναι σωστή στα επίπεδα 1 και 2 αλλά διαδικασίες ομογενοποίησης δημιουργούν μια νέα εγγραφή για αυτό το όργανο σε αυτή την ημερομηνία στο επίπεδο 3. Το επίπεδο και τα υπόλοιπα στοιχεία της εγγραφής περιλαμβάνονται -με κατάλληλο τρόπο- στο πεδίο των χαρακτηριστικών της εγγραφής, το οποίο συνεπώς πρέπει να είναι μέρος του συνόλου ταυτοποίησης της εγγραφής. Όλες οι εγγραφές όλων των επιπέδων ενός οργάνου περιλαμβάνονται στον ίδιο πίνακα (διαφορετικός για τα πρωτογενή και για τα δευτερογενή δεδομένα). Μια τιμή πχ. που ανήκει και στα 3 επίπεδα έχει μόνο μια εγγραφή με βάση τον κωδικό οργάνου, την ημερομηνία και τα χαρακτηριστικά, αν όμως για ένα όργανο και μια ημερομηνία υπάρχει μια τιμή στα επίπεδα 1 και 2 και μια άλλη στο 3, τότε για αυτό το ζεύγος υπάρχουν 2 εγγραφές που διαφέρουν μόνο στα χαρακτηριστικά. Τελικά δηλαδή προστίθεται νέα εγγραφή μόνο σε περίπτωση που η τιμή που πρέπει να προστεθεί είναι διαφορετική από αυτήν όλων των προηγούμενων επιπέδων, διαφορετικά αλλάζουν αντίστοιχα τα χαρακτηριστικά. Τα οφέλη από αυτή τη διαδικασία σε αποθηκευτικό χώρο και σε κυκλοφορία στο δίκτυο -σε σύγκριση με την "πλήρη" αποθήκευση όλων των επιπέδων- είναι προφανή.

3.4.3 Επιπλέον δεδομένα. Πρωτογενή δεδομένα επιπέδου 1 ενός πραγματικού οργάνου μπορεί να συγκεντρώσει και να αποθηκεύσει μόνο ο Φορέας - διαχειριστής του. Σε κάθε περίπτωση, τα "έπισημα" πρωτογενή και δευτερογενή δεδομένα οποιουδήποτε επιπέδου είναι αυτά που αποθηκεύει ο Φορέας - διαχειριστής. Ωστόσο, άλλοι Φορείς πάνω σε αυτά τα

δεδομένα μπορούν πιθανά επενδύσουν σημαντικό χρόνο και δυναμικό για να εφαρμόσουν είτε διαφορετικούς ελέγχους, είτε διαφορετικές τεχνικές συμπλήρωσης είτε ακόμα διαφορετικές μεθόδους συνάδρωσης -πιθανά σε άλλες χρονικές διακριτότητες-, οπότε να παράγουν είτε πρωτογενή επιπέδου 2, είτε επιπέδου 3 είτε διάφορα δευτερογενή - συναδρωσιμένα. Αυτά τα δεδομένα η ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ τους δίνει τη δυνατότητα να " σώσουν", δηλαδή να τα αποθηκεύσουν και να τα διαχειρισθούν ως δικές τους, "ανεπίσημες" χρονοσειρές. Με αυτό τον τρόπο μπορούν να υπάρχουν περισσότερες της μίας χρονοσειρές με ίδια χαρακτηριστικά από περισσότερους του ενός Φορείς. Τα δεδομένα αυτά αποθηκεύονται σε πίνακες με ονόματα ίδια με των "επίσημων" δεδομένων αλλά με το πρόθεμα x_ (από το "extra"). Πχ. για τον πίνακα raw_std2 (βλ. παρακάτω) που αποθηκεύει δεδομένα χρονοσειρών του οργάνου στον κόμβο του Φορέα - διαχειριστή, υπάρχει και ο αντίστοιχος πίνακας x_raw_std2 που αποθηκεύει δεδομένα του οργάνου με εναλλακτικό τρόπο παραγωγής σε κάποιο κόμβο -είτε διαφορετικό είτε ακόμα και ίδιο με τον διαχειριστικό (πχ. για δύο εναλλακτικές θεωρήσεις των δεδομένων). Υπάρχουν βέβαια και οι αντίστοιχες εγγραφές στο μητρώο χαρακτηριστικών σταδερών / χρονοσειρών. Αυτή η μεθοδολογία έχει το πρόβλημα πως, στην ΚΒΔ, μπορεί να υπάρχουν δεδομένα για το ίδιο όργανο την ίδια ημερομηνία και με τα ίδια χαρακτηριστικά σε περισσότερους του ενός κόμβους, δηλαδή τον "επίσημο", αυτόν του Φορέα - διαχειριστή, και κάποιους "ανεπίσημους", αυτούς που έκαναν κάποια επιπλέον / εναλλακτική επεξεργασία. Για να αποφευχθεί αυτό το γεγονός οι πίνακες δεδομένων x_ στην ΚΒΔ είναι απλές συνδέσεις προς άδειους πίνακες του κόμβου - συντονιστή. Δηλαδή, οι πίνακες x_ δεν είναι "ενεργοί" στην ΚΒΔ. Επίσης, κάθε κόμβος μπορεί να προσπελάσει μόνο τους "δικούς του" πίνακες επιπλέον δεδομένων.

3.4.4 Ταυτοποίηση χαρακτηριστικών χρονοσειρών - σταδερών. Σε κάθε περίπτωση, για κάθε χρονοσειρά υπάρχουν εγγραφές χαρακτηριστικών της στον κόμβο που την επεξεργάστηκε. Τα χαρακτηριστικά χρονοσειρών ταυτοποιούνται από τον κωδικό οργάνου στον οποίο αναφέρονται, το όνομα του Φορέα που είναι υπεύθυνος για την επεξεργασία της χρονοσειράς και έναν αύξοντα αριθμό χρονοσειράς για αυτό το όργανο και αυτό το Φορέα. Και αυτός ο αριθμός δεν χρειάζεται να εμφανίζεται στο χρήστη και βέβαια δεν πρέπει να δύναται να τροποποιηθεί από αυτόν.

Ακριβώς αντίστοιχη ταυτοποίηση υπάρχει και για τα χαρακτηριστικά σταδερών. Ταυτοποιούνται δηλαδή με βάση τα αντικείμενο στο οποίο αναφέρονται, το Φορέα που έκανε την επεξεργασία των δεδομένων και έναν αύξοντα αριθμό σταδερής για αυτό το αντικείμενο και αυτό το Φορέα.

3.4.5 Ταυτοποίηση σταθερών. Η ταυτοποίηση των σταθερών γίνεται με βάση το αντικείμενο στο οποίο αναφέρονται και τον αύξοντα αριθμό της σταθεράς που περιλαμβάνεται στα χαρακτηριστικά. Ο Φορέας δεν χρειάζεται διότι σε κάθε κόμβο αποθηκεύονται δεδομένα και σταθερές μόνο του Φορέα - διαχειριστή του κόμβου, συνεπώς εννοείται ο "τοπικός" Φορέας. Σε κάθε περίπτωση φαίνεται μια περισσότερο "στενή" σχέση των σταθερών με τα χαρακτηριστικά τους, μια και ο αύξων αριθμός των χαρακτηριστικών περιλαμβάνεται στην αποθήκευση των σταθερών. Αυτό γίνεται λόγω ακριβώς του στατικού χαρακτήρα των σταθερών. Αντίθετα, ενώ τα χαρακτηριστικά χρονοσειρών επίσης περιγράφουν την κάθε χρονοσειρά με ακρίβεια, εν τούτοις είναι δυνατό ο αύξων αριθμός χαρακτηριστικών να αλλάζει στη διάρκεια ζωής της χρονοσειράς.

3.4.6 Ταυτοποίηση γεγονότων. Τα γεγονότα ταυτοποιούνται με βάση το αντικείμενο στο οποίο αναφέρονται, την ημερομηνία και τον τύπο του γεγονότος. Αυτό βέβαια σημαίνει πως δεν υπάρχουν την ίδια ημερομηνία περισσότερα του ενός γεγονότα ίδιου τύπου για το ίδιο αντικείμενο. Η υπόθεση αυτή είναι ρεαλιστική.

3.5 Πίνακες αναφοράς και ονομασίες

Πολλές φορές μια πληροφορία στη ΒΔ πρέπει να περιλαμβάνεται μόνο σε ένα προκαθορισμένο σύνολο δυνατών τιμών (πεδίο ορισμού). Για παράδειγμα, ένα πεδίο για το "Νομό" στον οποίο ανήκει ένας σταθμός πρέπει να παίρνει τιμές μόνο μέσα από το σύνολο των Νομών της χώρας. Για να επιτευχθεί αυτός ο στόχος δημιουργείται ένας πίνακας που περιέχει όλους τους Νομούς της χώρας και κάθε εισαγωγή νέου σταθμού ή τροποποίηση υπάρχοντος "πυροδοτεί" έναν κανόνα { Γ } που ελέγχει την ύπαρξη του εισαχθέντος Νομού στον πίνακα των Νομών. Ο τελευταίος είναι έτσι ένα πίνακας αναφοράς -ή, σύμφωνα με άλλη ορολογία, είναι μέρος του Λεξικού Ορων-. Το πεδίο του αρχικού πίνακα που πρέπει να παίρνει τιμές μέσα από έναν πίνακα αναφοράς λέγεται εξωτερικό κλειδί { Γ }.

Στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ υπάρχουν δύο κύριοι τύποι εξωτερικών κλειδιών. Ο πρώτος είναι το αριθμητικό, δηλαδή ένας αριθμός - κώδικας μικρού μεγέθους (1 Byte). Χρησιμοποιείται σε όλες τις περιπτώσεις ανάγκης αποθήκευσης κωδίκων σε χρονοσειρές. Λόγω του μεγάλου όγκου των δεδομένων χρονοσειρών επιλέγεται η αριθμητική αποθήκευση για οικονομία χώρου. Για παράδειγμα ο καιρός αποθηκεύεται ως ένας κώδικας (σε πλήρη συμφωνία με αντίστοιχο κώδικα του WMO) του οποίου κάθε τιμή έχει μια ιδιαίτερη σημασία, όπως πχ. 1 = "Αίθριος", 2 = "Βροχή" κτλ., αντί να αποθηκεύεται η πλήρης περιγραφή. Το αλφαριθμητικό { Γ } εξωτερικό κλειδί, αυτό δηλαδή του οποίου η τιμή είναι μια συμβολοσειρά { Γ } 8 χαρακτήρων (Bytes) χρησιμοποιείται στις περιπτώσεις διαχειριστικών πληροφοριών, οπότε δίνει μια πληρέστερη εικόνα της τιμής του (πχ.

“ΑΤΤΙΚΗΣ” αντί για 15), χωρίς από την άλλη ιδιαίτερη επιβάρυνση σε αποθηκευτικό χώρο. Το μέγεθος των 8 χαρακτήρων είναι ένας καλός συμβιβασμός ανάμεσα στην σπατάλη χώρου και την ανάγκη πλήρους και ακριβούς περιγραφής της έννοιας. Χαρακτηριστικό των αλφαριθμητικών εξωτερικών κλειδιών είναι πως, αν στον πίνακα διαχειριστικών πληροφοριών πχ. `stuv` υπάρχει ένα εξωτερικό κλειδί με όνομα στήλης `{ Γ } abcd`, τότε υπάρχει ο πίνακας αναφοράς `abcds` (πληθυντικός) με κλειδί `abcd` από τις τιμές του οποίου πρέπει να παίρνει τιμή και το εξωτερικό κλειδί, δηλαδή να υπάρχει εγγραφή του `abcds, abcds .abcd`, τέτοια ώστε να ισχύει `stuv .abcd = abcds .abcd`.

Γενικά οι ονομασίες στη ΒΔ πρέπει να γίνονται με ελληνικούς άτονους χαρακτήρες. Η χρήση άτονων χαρακτήρων υπαγορεύεται από την ανάγκη σωστής ταξινόμησης και αποθήκευσης. Διαφορετικά, ένας τόνος που λείπει θα μπορούσε να κάνει την ίδια λέξη να αποθηκεύεται και να διαχειρίζεται από τη ΒΔ σε δύο διαφορετικές λέξεις, την άτονη και την τονισμένη. Πεζά και τονισμένα γράμματα μπορούν να χρησιμοποιούνται σε πεδία “κειμένου” όπου υπάρχουν παρατηρήσεις, σχόλια, αναφορές κοκ.

Ως προς τυχόν μίξη ελληνικών και λατινικών χαρακτήρων σε ονομασίες: στην περίπτωση που οι λατινικοί χαρακτήρες είναι ίδιοι με κάποιους ελληνικούς, έστω και όχι τους “αντίστοιχους” (πχ. τα ελληνικά “Τ” και “Ε” είναι τα ίδια αλλά και τα “αντίστοιχα” γράμματα με τα λατινικά “T” και “E” ενώ τα ελληνικά “Χ” και “Ρ” είναι τα ίδια γράμματα, αν και όχι “αντίστοιχα” με τα λατινικά “X” και “P”) τότε χρησιμοποιούνται πάντα οι ελληνικοί χαρακτήρες. Μόνο όταν δεν υπάρχουν ίδιοι ελληνικοί χαρακτήρες χρησιμοποιούνται οι λατινικοί, πχ. “D”, “R”.

3.6 Τύποι αντικειμένων και ΚΒΔ

Οι τύποι αντικειμένων της ΤΒΔ, σε σχέση με την ΚΒΔ, είναι οι ακόλουθοι:

3.6.1 Αναπαραγόμενα αντικείμενα τύπου `rpl` και `rpt`. Αντικείμενα του τύπου `rpl` (από το “replicate”) αποθηκεύουν μόνο τοπικές (δηλαδή διαχειριστής τους είναι ο “τοπικός” Φορέας) πρωτεύουσες (αρχής γενομένης από το σταθμό και στη συνέχεια όλα τα συνδεδεμένα όργανα, χρονοσειρές κτλ.) διαχειριστικές πληροφορίες οι οποίες τελικά αντιγράφονται σε όλους τους άλλους κόμβους. Τα περιεχόμενα των αντικειμένων αυτού του τύπου ακολουθούν την ακόλουθη “πορεία”: μια νέα εγγραφή εισάγεται καταρχήν στον πίνακα τύπου `rpl` με όνομα `xxx_rpl` για αντικείμενα τύπου `xxx` όπου `xxx` μπορεί να είναι διαχειριστικές πληροφορίες, δηλαδή `stations` (σταθμοί), `instruments` (όργανα), `timeseries` (χαρακτηριστικά χρονοσειρών), `constants` (χαρακτηριστικά σταθερών) ή `events` (γεγονότα). Κατάλληλοι κανόνες της ΒΔ εισάγουν αυτόματα -χωρίς γνώση του χρήστη- τη νέα εγγραφή και σε έναν άλλον πίνακα με όνομα `xxx_rpl_ins`. Αντίστοιχα,

αν πρόκειται για διαγραφή εγγραφής κατάλληλοι κανόνες εισάγουν την ταυτότητα της εγγραφής που διαγράφηκε (πχ. κωδικός ΥΔΡΟΣΚΟΠΙΟΥ) σε έναν πίνακα `xxx_rpl_del`. Η τροποποίηση κάποιας εγγραφής αντιμετωπίζεται σα διαγραφή και στη συνέχεια επανεισαγωγή της. Με λίγα λόγια, οι πίνακες `_rpl_ins` και `_rpl_del` κρατούν μια εικόνα των τροποποιήσεων που έγιναν στις τοπικές διαχειριστικές πληροφορίες κατά τη διάρκεια της ημέρας. Κατά τη διάρκεια της νύχτας, οπότε και ο φόρτος του δικτύου είναι μικρός, ένας "δαίμονας" { Γ } του συστήματος, ο `hrpld` (HYDROSCOPE replication daemon) διαβάζει τις πληροφορίες που υπάρχουν σε όλους τους `_rpl_ins` και `_rpl_del` πίνακες της ΤΒΔ και με διαδοχικές συνδέσεις με όλες τις άλλες (απομακρυσμένες) ΤΒΔ αντιγράφει τις πληροφορίες αυτές, δηλαδή διαγράφει τις πληροφορίες που ταυτοποιούνται από τα περιεχόμενα του πίνακα `_rpl_del` και εισάγει αυτές που περιέχονται στον πίνακα `_rpl_ins`, στα αντίστοιχα απομακρυσμένα αντικείμενα τύπου `rmt`, δηλαδή `xxx_rmt` όπου `xxx` όπως παραπάνω. Με αυτό τον τρόπο την επόμενη ημέρα όλες οι διαχειριστικές πληροφορίες όλων των κόμβων είναι ενημερωμένες στους πίνακες `rmt`.

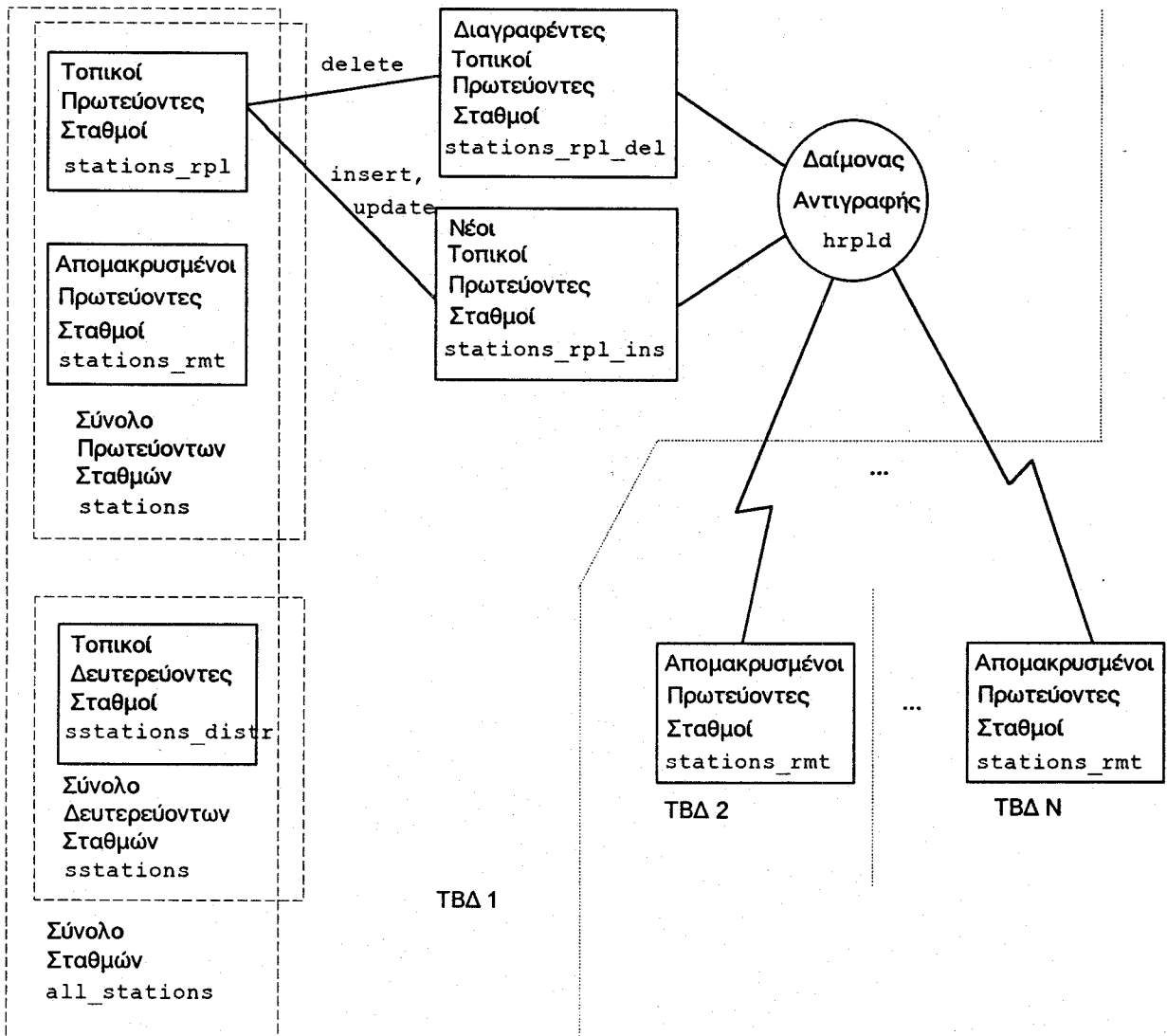
Τα αντικείμενα τύπου `rmt` (`stations_rmt`, `events_rmt` κτλ.) (από το "remote") αποθηκεύουν τις πρωτεύουσες πληροφορίες των υπόλοιπων κόμβων του δικτύου που έχουν αντιγραφεί μέχρι και την προηγούμενη νύχτα στον τοπικό κόμβο. Τα αντικείμενα αυτά είναι και τα μόνα αντικείμενα της ΤΒΔ στα οποία έχουν δικαίωμα εγγραφής οι απομακρυσμένοι κόμβοι. Έχουν φυσικά την ίδια μορφή με τα αντίστοιχα `rpl` αντικείμενα.

Η ανάκτηση διαχειριστικών πληροφοριών κατηγορίας `xxx` (όπου `xxx` όπως παραπάνω) πρέπει πάντα να γίνεται από την όγη `xxx` που είναι η λογική ένωση των πληροφοριών των πινάκων τύπου `rpl` και `rmt`, δηλαδή η λογική ένωση των διαχειριστικών πληροφοριών του τοπικού κόμβου και όλων των απομακρυσμένων, άρα τελικά η λογική ένωση όλων των πρωτεύουσών διαχειριστικών πληροφοριών της ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ. Στην όγη δεν μπορεί να γίνει εισαγωγή - τροποποίηση πληροφοριών. Μόνο σε αυτή την περίπτωση χρησιμοποιείται ο υποκείμενος πίνακας τύπου `rpl`, για εισαγωγή δηλαδή και τροποποίηση των τοπικών πρωτεύουσών πληροφοριών. Στον υποκείμενο πίνακα τύπου `rmt` δεν μπορούν να γράγουν οι "τοπικές" εφαρμογές παρά μόνο οι απομακρυσμένοι δαίμονες `hrpld`.

Στην ΚΒΔ, μια και έχει το ίδιο σχήμα με την ΤΒΔ, υπάρχει επίσης μια όγη `xxx` για τις διαχειριστικές πληροφορίες της κατηγορίας `xxx`. Η όγη αυτή πρέπει να περιέχει αντίστοιχα τη λογική ένωση όλων των πρωτεύουσών διαχειριστικών πληροφοριών `xxx` της ΚΒΔ. Μια και οι κατά τόπους πίνακες `rpl` περιέχουν τις τοπικές πρωτεύουσες διαχειριστικές πληροφορίες, η λογική ένωση όλων των πινάκων `rpl` είναι η ζητούμενη όγη. Υπάρχει βέβαια μια διαφορά: ενώ η όγη `xxx` της ΤΒΔ είναι η ένωση των "τωρινών" τοπικών πρωτεύουσών διαχειριστικών πληροφοριών (`rpl`) `xxx` με τις "χτεσινές" απομακρυσμένες πρωτεύουσες διαχειριστικές πληροφορίες `xxx`, η όγη `xxx` της ΚΒΔ, ως λογική ένωση των πινάκων `xxx_rpl`, είναι η ένωση των "τωρινών" τοπικών πρωτεύουσών διαχειριστικών

πληροφοριών. Τις πλέον πρόσφατες πληροφορίες έχει λοιπόν η ΚΒΔ. Αυτό δεν μπορεί να αποφευχθεί και οφείλεται προφανώς στο ό,τι η ενημέρωση των πινάκων rmt γίνεται το βράδυ, για καλύτερη κατανομή του φορτίου του δικτύου.

3.6.2 Τοπικά αντικείμενα τύπου 1c1. Οι πληροφορίες αυτού του τύπου είναι βοηθητικές πληροφορίες της ΒΔ. Πρόκειται δηλαδή για πληροφορίες που είναι κοινές σε όλους τους κόμβους. Κατά συνέπεια, οι πληροφορίες αυτού του τύπου στην ΤΒΔ προσπελούνται από τους αντίστοιχους πίνακες, στη δε ΚΒΔ οι πίνακες τύπου 1c1 (από το 'local') είναι απευθείας συνδεδεμένοι { Γ } με τους αντίστοιχους πίνακες της ΤΒΔ του κόμβου - συντονιστή. Με αυτό τον τρόπο κάθε ερώτηση προς αντικείμενο τύπου 1c1 στην ΚΒΔ αποστέλλεται στην ΤΒΔ του κόμβου - συντονιστή και απαντάται με βάση τις πληροφορίες του αντίστοιχου αντικειμένου της.



Σχήμα 3.6.1: Τύποι αντικειμένων ΚΒΔ και εφαρμογή στην περίπτωση των σταθμών

3.6.3 Τοπικά αντικείμενα τύπου lc11. Οι πληροφορίες τύπου lc11 είναι βοηθητικές πληροφορίες οι οποίες όμως δεν έχουν "νόημα" στην ΚΒΔ. Για παράδειγμα, πληροφορίες σχετικές με ονόματα χρηστών δεν μπορούν να χρησιμοποιηθούν στην ΚΒΔ διότι αναφέρονται πάντα σε έναν μόνο από τους συμμετέχοντες κόμβους. Στην περίπτωση αυτή, για lc11 αντικείμενο με όνομα γγγ στην ΤΒΔ, υπάρχει και ένα κενό αντικείμενο με όνομα γγγ_1 στην ΤΒΔ του κόμβου - συντονιστή. Το αντικείμενο της ΚΒΔ είναι απευθείας συσχετισμένο με αυτό το κενό αντικείμενο. Με τον τρόπο αυτό το σχήμα της ΒΔ μένει ίδιο τόσο στην τοπική όσο και στην κατανεμημένη μορφή του αλλά δεν προσπελαύνονται ανακριβείς πληροφορίες.

3.6.4 Μη αναπαραγόμενα αντικείμενα τύπου distr. Όλες οι πληροφορίες των οποίων η λογική ένωση πρέπει να προσπελαύνεται μέσα από την ΚΒΔ, εκτός από τα τύπου rpl, ανήκουν στον τύπο distr. Αυτό σημαίνει πως το αντίστοιχο αντικείμενο της ΚΒΔ είναι μια όγη που αποτελεί τη λογική ένωση των πληροφοριών των αντικειμένων όλων των συμμετεχόντων ΤΒΔ και άρα περιλαμβάνει το σύνολο των σχετικών πληροφοριών του ΥΔΡΟΣΚΟΠΙΟΥ. Όλα τα αντικείμενα τα σχετικά με χρονοσειρές και σταθερές είναι αυτού του τύπου, αποθηκεύονται δηλαδή μόνο στον κόμβο / ΤΒΔ του Φορέα - διαχειριστή και προσπελαύνονται μέσω της ΚΒΔ. Επίσης, από τις διαχειριστικές πληροφορίες, αυτές που αφορούν στους δευτερεύοντες σταθμούς (και τα αντίστοιχα όργανα, χαρακτηριστικά χρονοσειρών και σταθερών και γεγονότα) και δεν αντιγράφονται σε όλους τους κόμβους, είναι τύπου distr.

3.7 Αποθήκευση χρονοσειρών

Οι χρονοσειρές αποτελούν τη μεγάλη πλειοψηφία των πληροφοριών που αποθηκεύονται στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ. Η αποθήκευσή τους υπόκειται σε ορισμένες αρχές που περιγράφονται στη συνέχεια.

3.7.1 Ακέραιες τιμές. Όλες οι τιμές χρονοσειρών που αποθηκεύονται στη ΒΔ είναι ακέραιοι αριθμοί. Οι πραγματικές τιμές που μετρώνται είναι βέβαια πραγματικοί αριθμοί. Ωστόσο, για να αποθηκευθεί ένας πραγματικός αριθμός κινητής υποδιαστολής { Γ } απαιτούνται τουλάχιστον 4 Bytes χώρου. Με αυτό τον τρόπο η ζητούμενη ακρίβεια φτάνει τα 8 δεκαδικά ψηφία και η μέγιστη δυνατή τιμή το 10^{38} [INGRES91a]. Σε καμία όμως περίπτωση δεν απαιτείται τόσο μεγάλη ακρίβεια (0, 1, 2 και σε εξαιρετικές περιπτώσεις 3 δεκαδικά ψηφία είναι η συνηθής απαιτούμενη ακρίβεια) ούτε και τόσο μεγάλες τιμές. Κατά συνέπεια, η αποθήκευση στη ΒΔ πραγματικών αριθμών είναι σπατάλη χώρου. Για το λόγο

αυτό χρησιμοποιούνται ακέραιες τιμές μεγέθους 1, 2 ή 4 Bytes, ανάλογα με την ακρίβεια και τις ακραίες τιμές της μεταβλητής. Η πραγματική τιμή βρίσκεται διαιρώντας, στο επίπεδο της εφαρμογής, την αποθηκευμένη τιμή με $10^{\text{ακρίβεια}}$. Για παράδειγμα, μια ακέραια τιμή 2 Bytes μπορεί να πάρει τιμή μέχρι 32767. Αυτό σημαίνει πως με ακρίβεια 0 δεκαδικών μπορούν να αποθηκευτούν πραγματικές τιμές μέχρι 32767 ($= 32767 / 10^0$), με ακρίβεια 1 μέχρι 3276.7 ($= 32767 / 10^1$), με ακρίβεια 2 μέχρι 327.67 ($= 32767 / 10^2$) και με ακρίβεια 3 μέχρι 32.767 ($= 32767 / 10^3$). Τα οφέλη από αυτή την τακτική είναι σημαντικά, μια και οι περισσότερες από τις τιμές που αποθηκεύονται στη ΒΔ "χωρούν", από άποψη μέγιστης τιμής και ακρίβειας, σε 1 ή 2 Bytes. Ορισμένες μόνο εγγραφές χρειάζεται να χρησιμοποιήσουν και τα 4 Bytes ακέραιας τιμής, τα οποία θα χρησιμοποιούσαν έτσι και αλλιώς όλες σε περίπτωση αποθήκευσης πραγματικών αριθμών. Έτσι γίνεται σοβαρή οικονομία τόσο σε αποθηκευτικό χώρο στο δίσκο όσο και σε κυκλοφορία πάνω από το δίκτυο. Το μειονέκτημα είναι η επιπλέον απαιτούμενη διαίρεση προκειμένου για ανάκτηση δεδομένων ή πολλαπλασιασμός για εισαγωγή δεδομένων που πρέπει να γίνει από την εφαρμογή. Με το διαθέσιμο ισχυρό υλικό { Γ } πράξεων κινητής υποδιαστολής αυτό το μειονέκτημα είναι καθαρά θεωρητικό και δεν επιβαρύνει αισθητά την απόδοση των εφαρμογών.

3.7.2 Κανονική μορφή αποθήκευσης. Αναγκαίο χαρακτηριστικό της ΒΔ πρέπει να είναι η δυνατότητα αποθήκευσης των χρονοσειρών όλων των δυνατών οργάνων του ΥΔΡΟΣΚΟΠΙΟΥ. Το πρόβλημα προκύπτει από το γεγονός πως ένα όργανο μπορεί ανά πάσα χρονική στιγμή να μετρά μια τιμή που αποτελείται από δύο ή περισσότερα φυσικά μεγέθη. Για παράδειγμα ένα θερμόμετρο μετρά τη θερμοκρασία, ένα ανεμόμετρο όμως μετρά και διεύθυνση και ταχύτητα ανέμου και ένα όργανο νέφωσης "μετρά" ταυτόχρονα 9 διαφορετικά στοιχεία. Συνεπώς η αποθήκευση των χρονοσειρών πρέπει να ικανοποιεί αυτές τις διαφορετικές απαιτήσεις. Για το σκοπό αυτό υπάρχουν διαφορετικές μορφές αποθήκευσης των τιμών των οργάνων.

Στην κανονική μορφή αποθήκευσης { Γ }, std, κάθε τιμή που μετράται από το όργανο αποθηκεύεται ως ένα ή περισσότερα ακέραια πεδία, ως εξής:

$$\langle \text{τιμή} \rangle = \langle \text{πεδίο } 1 \rangle [\langle \text{πεδίο } -2 \rangle \dots \langle \text{πεδίο } -\kappa \rangle]$$

Η κανονική μορφή αποθήκευσης αναλύεται περαιτέρω σε περιπτώσεις, ανάλογα με την τιμή του κ , τον αριθμό δηλαδή των πεδίων και το εύρος τους, σε Bytes αποθήκευσης ακέραιων τιμών (1, 2 ή 4). Οι περιπτώσεις αυτές χαρακτηρίζονται με ένα όνομα, της μορφής stdxxx, όπου xxx χαρακτηριστικό της μορφής. Έτσι υπάρχουν οι μορφές αποθήκευσης std2 (1 πεδίο - κ = 1- 2 Bytes), std11 (2 πεδία - κ = 2- από 1 Byte το καθένα), std211 (3 πεδία - κ = 3-, το πρώτο 2 Bytes και τα άλλα 2 από 1 Byte) κ.ο.κ. Οι μορφές προέκυψαν από προσεκτική μελέτη όλων των οργάνων και καθορισμό του τρόπου με τον οποίο πρέπει να αποθηκεύουν τις τιμές τους.

Τα πεδία ονομάζονται με τον ίδιο τρόπο:

`value0 [value1...value (κ-1)]`

ανεξάρτητα από το εύρος τους. Κατά συνέπεια, οι εφαρμογές δεν χρειάζεται να διαφοροποιούν την αντιμετώπιση των πεδίων ανάλογα με το εύρος τους, μια και έχουν όλα κοινή ονοματολογία και υπόκεινται όλα στην ίδια διαδικασία διαίρεσης για την παρουσίαση των πραγματικών τιμών.

3.7.3 Αποθήκευση πρωτογενών. Κάθε πρωτογενής (είτε από πραγματικό είτε από παράγωγο όργανο) τιμή χρονοσειράς που αποθηκεύεται στη ΒΔ με κάποια κανονική μορφή αποθήκευσης (με ένα ή περισσότερα πεδία) αναφέρεται στο όργανο που μέτρησε την τιμή, μια ημερομηνία μέτρησης και ένα σύνολο χαρακτηριστικών. Το όνομα του πίνακα που αποθηκεύει τη χρονοσειρά αρχίζει από `raw` και το δεύτερο συνθετικό του ονόματος είναι το όνομα της κανονικής μορφής αποθήκευσης, `stdxxx`. Τα δύο συνθετικά ενώνονται με το χαρακτήρα "_", έτσι ώστε το τελικό όνομα των πινάκων αποθήκευσης πρωτογενών να είναι `raw_stdxxx`, όπου `stdxxx` είναι το όνομα της μορφής αποθήκευσης, πχ. `std2`, `std11`, `std211` κοκ. Οι πίνακες αυτοί είναι φυσικά τύπου `distr`, δεν αντιγράφονται δηλαδή και στην ΚΒΔ υπάρχει με το ίδιο όνομα η λογική ένωση όλων των αντίστοιχων πινάκων των συμμετεχουσών ΤΒΔ. Τα πεδία των πινάκων `raw` είναι, εκτός αυτών που απορρέουν από τη μορφή αποθήκευσης (`value0` και ίσως `value1`, `value2` κτλ.) τα ακόλουθα:

3.7.3.1 Κωδικός οργάνου. Ο κωδικός οργάνου είναι ένα πεδίο τύπου `hsiid` με όνομα `instrument`.

3.7.3.2 Ημερομηνία. Η ημερομηνία είναι ένα πεδίο τύπου `hdate` με όνομα `date`. Ο τύπος `hdate` είναι τύπος δεδομένων οριζόμενος από το χρήστη και στην πραγματικότητα είναι ένας ακέραιος 4 Bytes που αποθηκεύει τον αριθμό των πρώτων λεπτών που έχουν περάσει από μια αρχική συμβατική ημερομηνία, την 1/1/1 00:00. Αυτό δε φαίνεται στο χρήστη, ο οποίος βλέπει την κανονική γνώριμη μορφή αναπαράστασης ημερομηνιών. Ο τύπος αυτός δεδομένων έχει, όπως είναι φυσικό, χρονική διακριτότητα ενός πρώτου λεπτού, και αυτή είναι και η μέγιστη δυνατή χρονική διακριτότητα στο σύστημα του ΥΔΡΟΣΚΟΠΙΟΥ. Η διακριτότητα αυτή κρίνεται επαρκής για τον τύπο των αποθηκευόμενων δεδομένων. Το μεγάλο πλεονέκτημα του τύπου δεδομένων `hdate` είναι πως, με μια μικρή και χωρίς δυσμενείς επιπτώσεις στη λειτουργικότητα του συστήματος παραχώρηση σε διακριτότητα σε σχέση με τον ενδοκτισμένο { Γ } τύπο `date` (πρώτα λεπτά αντί δευτερόλεπτα), επιτυγχάνει πολύ σοβαρή οικονομία χώρου και κυκλοφορίας πάνω από το δίκτυο, μια και ο ενδοκτισμένος τύπος καταλαμβάνει 12 Bytes χώρου αντί για τα 4 του οριζόμενου από το χρήστη. Στην περίπτωση των πρωτογενών χρονοσειρών

στο πεδίο της ημερομηνίας αποθηκεύεται η ακριβής ημερομηνία, ώρα και πρώτο λεπτό που έγινε η μέτρηση. Σε περίπτωση που αυτή η σκριβής τιμή δεν είναι γνωστή ή δεν έχει νόημα (πχ. στην περίπτωση παράγωγων οργάνων), τότε αποθηκεύεται η ημερομηνία στην οποία αναφέρεται η μέτρηση και για την ώρα και τα πρώτα λεπτά ακολουθείται μια σύμβαση, ανάλογα με το είδος του οργάνου και της χρονοσειράς. Σημασία στην περίπτωση αυτή δεν έχει τόσο η απόλυτη τιμή που χρησιμοποιείται για την ώρα και τα λεπτά όσο να είναι αυτή η τιμή ίδια καθ' όλη τη διάρκεια της χρονοσειράς ή ακόμα καλύτερα για όλα τα όργανα του ίδιου τύπου. Για παράδειγμα, τα βροχόμετρα μπορούν να αποθηκεύονται στις 08:00, τα θερμόμετρα στις 14:00 κοκ. Σημειωτέον πως οι μετρήσεις μετεωρολογικών μεταβλητών αναφέρονται τις περισσότερες φορές σε ώρα GMT.

3.7.3.3 Χαρακτηριστικά. Η λέξη κατάστασης { Γ } είναι αυτή που καθορίζει το είδος, τα χαρακτηριστικά και την ποιότητα της αποθηκευμένης πρωτογενούς τιμής. Πρόκειται για έναν ακέραιο 4 Bytes με όνομα πεδίου status που με κατάλληλη διάταξη των bits του, όπως περιγράφεται στη συνέχεια, χαρακτηρίζει την εγγραφή.

Σύμφωνα με τα παραπάνω, τα πρωτογενή δεδομένα αποθηκεύονται βασικά σε πίνακες με όνομα του τύπου raw_stdxxx, όπου stdxxx η ονομασία της κανονικής μορφής αποθήκευσης. Τα πεδία των πινάκων αυτών είναι τα ακόλουθα:

```
instrument date status value0 [value1...valueλ]
```

όπου $\lambda = \kappa - 1$.

3.7.4 Λέξη κατάστασης. Η λέξη κατάστασης είναι ακέραιος 4 Bytes (32 bits). Ο ακέραιος αυτός είναι λογικά χωρισμένος σε πεδία, όπου το κάθε πεδίο αποτελείται από ένα ή περισσότερα bits. Κάθε ένα από αυτά τα πεδία χαρακτηρίζεται - ονομάζεται από ένα κεφαλαίο λατινικό γράμμα (A - Z) με βάση το οποίο γίνεται η αναφορά στο πεδίο. Ανάλογα με το εύρος του πεδίου μπορούν σε αυτό να αποθηκευθούν μικρές ακέραιες τιμές από 0 ως $2^{\text{«εύρος πεδίου»} - 1}$. Για παράδειγμα, για εύρος πεδίου 3 bits οι δυνατές τιμές είναι από 0 ως 7 ($= 2^3 - 1$). Κάθε τέτοιο πεδίο bits { Γ } αντιστοιχίζεται σε κάποιο από τα δυνατά χαρακτηριστικά της εγγραφής - σημαίες { Γ }. Η τιμή του πεδίου είναι και η τιμή αυτού του χαρακτηριστικού για τη συγκεκριμένη εγγραφή. Για παράδειγμα, έστω πως ένα δυνατό χαρακτηριστικό μιας εγγραφής είναι το αν έχει υποστεί έλεγχο ακραίων τιμών και αν ναι, ποιο ήταν το αποτέλεσμα του [Ava92]. Αν το πεδίο R που αντιστοιχίζεται στον έλεγχο ακραίων τιμών για τη λέξη κατάστασης αυτής της εγγραφής έχει τιμή 0 αυτό σημαίνει πως δεν έχει γίνει έλεγχος ακραίων τιμών, αν έχει τιμή 1 πως έχει γίνει και έχει πετύχει, αν έχει τιμή 2 πως έχει γίνει και έχει αποτύχει κοκ. Άλλο δυνατό χαρακτηριστικό της λέξης κατάστασης είναι το επίπεδο της εγγραφής. Έτσι υπάρχουν 3 πεδία με ονόματα K, L, M που

αντιστοιχούν στα επίπεδα 1, 2 και 3 αντίστοιχα. Έτσι, αν για μια εγγραφή κάποιο από αυτά τα πεδία στη λέξη κατάστασης έχει τιμή 1 τότε αυτό σημαίνει πως η εγγραφή ανήκει στο αντίστοιχο επίπεδο, αν έχει τιμή 0 δεν ανήκει. Με αυτό το "συμπαγή" τρόπο είναι δυνατό να καθορισθούν όλα τα χαρακτηριστικά μιας εγγραφής, οι έλεγχοι που έχει υποστεί, οι συνθήκες υπόβαθρου { Γ } κατά τη στιγμή της μέτρησης (αυτές όπως είναι φυσικό εισάγονται μαζί με τα πρωτογενή δεδομένα επιπέδου 1), το επίπεδο στο οποίο ανήκει κοκ.

Στον ακόλουθο πίνακα φαίνονται όλα τα πεδία της λέξης κατάστασης.

Θέση (bit)	Πεδίο bits	Ερμηνεία	Εύρος (bits)	Σημασία τιμών πεδίων
0	L	Επίπεδο 1	1	1 = εγγραφή ανήκει στο επίπεδο 1
1	M	Επίπεδο 2	1	1 = εγγραφή ανήκει στο επίπεδο 2
2	N	Επίπεδο 3	1	1 = εγγραφή ανήκει στο επίπεδο 3
3	E	Εκτίμηση	1	1 = τιμή εγγραφής κατ' εκτίμηση
4	A	Αλφαριθμητικό σχόλιο	1	1 = υπάρχει στον πίνακα <code>events_rpl</code> αλφαριθμητικό σχόλιο για αυτή την εγγραφή (με κλειδιά όργανο - ημερομηνία και το όνομα του πίνακα)
5	J	Υποπτη τιμή	1	1 = η τιμή της εγγραφής είναι ύποπτη
6	Q	Δείκτης ποιότητας	2	0 = δεν υπάρχει ή εγγραφή ορδή 1 = εγγραφή σχεδόν ορδή 2 = εγγραφή σχεδόν λάθος 3 = εγγραφή λάθος
8	G	Συμπλήρωση	1	1 = η εγγραφή είναι αποτέλεσμα συμπλήρωσης

9	H	Ομογενοποίηση	2	0 = δεν έγινε ομογενοποίηση 1 = έγινε ομογενοποίηση και πέτυχε 2 = έγινε ομογενοποίηση και απέτυχε αλλά δεν έγινε ανόρθωση 3 = έγινε ομογενοποίηση και απέτυχε και έγινε ανόρθωση
11	U	Πεδίο χρήστη	3	Ελεύθερο πεδίο για χρήση και ερμηνεία από διάφορες εφαρμογές και για διάφορα είδη πληροφορίας (τιμές 0 - 7)
14	V	Στάθμη - παροχή	2	Ειδικά για καμπύλες στάθμης - παροχής 0 = σημείο δεν έχει ελεγχθεί 1 = σημείο ανήκει σε καμπύλη 2 = σημείο δεν ανήκει σε καμπύλη
16	-	Ελεύθερο πεδίο	1	Ελεύθερο πεδίο για χρήση και ερμηνεία από διάφορες εφαρμογές. Διαδέσιμα χαρακτηριστικά γράμματα: B, K, P, O, Y, W.
17	R	Ελεγχος ακραίων τιμών	3	0 = δεν έγινε 1 = έγινε και πέτυχε 2 = έγινε, απέτυχε και τέθηκε η μεγαλύτερη δυνατή τιμή (G) 3 = έγινε, απέτυχε και τέθηκε η μικρότερη δυνατή τιμή (L) 4 = έγινε, απέτυχε αλλά η τιμή της εγγραφής διατηρήθηκε (V)
20	X	Ελεγχος χωρικής συνέπειας	2	0 = δεν έγινε 1 = έγινε και πέτυχε 2 = έγινε και απέτυχε
22	C	Ελεγχος εσωτερικής συνέπειας	2	0 = δεν έγινε 1 = έγινε και πέτυχε 2 = έγινε και απέτυχε

24	T	Έλεγχος χρονικής συνέπειας	3	0 = δεν έγινε 1 = έγινε και πέτυχε 2 = έγινε, απέτυχε και τέθηκε η μεγαλύτερη δυνατή τιμή (G) 3 = έγινε, απέτυχε και τέθηκε η μικρότερη δυνατή τιμή (L) 4 = έγινε, απέτυχε αλλά η τιμή της εγγραφής διατηρήθηκε (V)
27	S	Χιόνι	1	1 = Κατά τη στιγμή της μέτρησης υπήρχε χιόνι
28	I	Πάγος	1	1 = Κατά τη στιγμή της μέτρησης υπήρχε πάγος
29	F	Παγετός	1	1 = Κατά τη στιγμή της μέτρησης υπήρχε παγετός
30	D	Κατάδυση	1	1 = Κατά τη στιγμή της μέτρησης ο σταθμός ήταν καταδυμένος (πλημμύρα)
31	Z	Μηδέν	1	Δεσμευμένο πεδίο προσήμου με τιμή πάντα 0 ώστε η λέξη κατάστασης να είναι πάντα θετική

Πίνακας 3.7.4.1: Η μορφή της λέξης κατάστασης και η σημασία των πεδίων της

Ο χειρισμός της λέξης κατάστασης γίνεται μέσα από δύο κατάλληλες συναρτήσεις SQL ορισμένες από το χρήστη, τις `getstatus()` και `setstatus()`. Η `getstatus(status, fields)` επιστρέφει την τιμή του πεδίου ή των πεδίων `bits fields` της λέξης κατάστασης `status`, π.χ. η ερώτηση SQL `select ... from ... where getstatus(status, 'L') = 1 and ...` επιστρέφει τις εγγραφές που, εκτός των άλλων συνθηκών, ανήκουν στο επίπεδο 1, αυτές δηλαδή που έχουν το πεδίο `bits L = 1`. Η `setstatus(status, fields_values)` αλλάζει στη λέξη κατάστασης `status` την τιμή του πεδίου ή των πεδίων `bits` σύμφωνα με τις τιμές που περιέχονται στο όρισμα `fields_values` και επιστρέφει τη νέα λέξη κατάστασης. Π.χ. `setstatus(status, 'L1')` αλλάζει τη λέξη κατάστασης `status` ώστε να γίνει το πεδίο `bits L` ίσο με 1. Η ερώτηση SQL `update ... set status = setstatus(status, 'M1') where getstatus(status, 'R')`

= 2 θέτει την τιμή του status έτσι ώστε το πεδίο bits M να πάρει την τιμή 1 στις περιπτώσεις για τις οποίες το πεδίο bits R έχει τιμή 2. Η ακριβής χρήση των συναρτήσεων αυτών αλλά και οι περιορισμοί που απορρέουν από τη χρήση τους πάνω από την ΚΒΔ αναλύονται εκτενώς στο αντίστοιχο κεφάλαιο.

Η υλοποίηση των επιπέδων μέσω της λέξης κατάστασης έχει σοβαρές θετικές συνέπειες στον αποθηκευτικό χώρο, μια και επιτρέπει η ίδια φυσική εγγραφή να αναφέρεται σε περισσότερα από ένα επίπεδα. Επίσης η λέξη κατάστασης είναι αυτή που επιτρέπει την αποτελεσματική, οικονομική και γρήγορη διαχείριση των χαρακτηριστικών μιας εγγραφής και τελικά δίνει κάποιου είδους "ταυτότητα" σε αυτήν. Για τους λόγους αυτούς είναι απαραίτητο οι εφαρμογές να λαμβάνουν υπόψη τους, να επεξεργάζονται και να διαχειρίζονται τις λέξεις κατάστασης των εγγραφών. Βέβαια η λέξη κατάστασης περιπλέκει την εισαγωγή, τροποποίηση και διαγραφή τιμών: η εισαγωγή μιας τιμής σε ένα επίπεδο δεν σημαίνει απλά εισαγωγή μιας νέας εγγραφής αλλά έλεγχο για το αν ήδη η εγγραφή υπάρχει σε κάποιο προηγούμενο επίπεδο για το ίδιο ζεύγος οργάνου - ημερομηνίας και θέση του αντίστοιχου πεδίου bits, σε περίπτωση που ο έλεγχος αποβεί θετικός, διαφορετικά η εισαγωγή μιας νέας εγγραφής αλλά με παράλληλη αντιγραφή των υπόλοιπων (εκτός του επιπέδου) χαρακτηριστικών που περιέχονται στη λέξη κατάστασης. Παρόμοιες δυσκολίες παρουσιάζονται σε επίπεδο εφαρμογής και στις άλλες ενέργειες.

3.7.5 Αποθήκευση μορφής λίστας. Υπάρχουν ορισμένες περιπτώσεις στις οποίες οι χρονοσειρές για μια σημαντική χρονική περίοδο έχουν σταθερή περίοδο μέτρησης έτσι ώστε η συνεχής αποθήκευση των ημερομηνιών της κανονικής μορφής αποθήκευσης να είναι αντιοικονομική. Σε αυτές τις περιπτώσεις θα αρκούσε η αποθήκευση μιας μόνο ημερομηνίας έναρξης και του βήματος (χρονικής διακριτότητας) της χρονοσειράς ώστε να αποθηκευθεί με περισσότερο οικονομικό τρόπο. Αυτή την ανάγκη καλύπτει η αποθήκευση μορφής λίστας, 1stxxx. Για το xxx ισχύουν όσα και για την κανονική μορφή, stdxxx. Η μορφή 1stxxx είναι μια εναλλακτική μορφή αποθήκευσης της κανονικής μορφής stdxxx ιδιαίτερα κατάλληλη για χρονοσειρές πρωτογενών με σταθερό χρονικό βήμα. Έχει τη γενική διάταξη:

<όργανο><ημερομηνία><χαρακτηριστικά><αριθμός><διάστημα>
<τιμή1>...<τιμή32>

Στο πεδίο <όργανο> αποθηκεύεται ο κωδικός ΥΔΡΟΣΚΟΠΙΟΥ του οργάνου που έκανε το σύνολο των μετρήσεων. Η ημερομηνία - ώρα - πρώτο λεπτό που έγινε η πρώτη μέτρηση, αυτή δηλαδή που αποθηκεύεται στο πεδίο <τιμή1>, είναι αυτή που αποθηκεύεται στο πεδίο <ημερομηνία>. Κάθε εγγραφή της μορφής λίστας μπορεί να αποθηκεύσει μέχρι 32 τιμές. Ο πραγματικός αριθμός των αποθηκευμένων τιμών υπάρχει στο πεδίο <αριθμός> ενώ στο πεδίο <διάστημα> υπάρχει ο αριθμός των πρώτων λεπτών που μεσολαβούν από

μια μέτρηση μέχρι την αμέσως επόμενη της. Κατά συνέπεια η i -οστή μέτρηση, όπου $1 \leq i \leq \langle \text{αριθμός} \rangle$ αναφέρεται στην ημερομηνία $\langle \text{ημερομηνία} \rangle + (i - 1) * \langle \text{διάστημα} \rangle$. Τέλος, στο πεδίο $\langle \text{χαρακτηριστικά} \rangle$, το οποίο είναι μια λέξη κατάστασης, αποθηκεύονται τα χαρακτηριστικά των εγγραφών. Τα χαρακτηριστικά αυτά πρέπει να είναι κοινά για όλες τις αποθηκευμένες εγγραφές, δηλαδή να ανήκουν όλες στο ίδιο επίπεδο, να έχουν όλες υποστεί τους ίδιους ελέγχους με ίδια αποτελέσματα κοκ. Είναι προφανές πως κάτι τέτοιο περιορίζει τη χρήση της μορφής λίστας σε περιπτώσεις μεγάλης ομοιογένειας των δεδομένων, τόσο από άποψη διαστήματος - βήματος μέτρησης όσο και χαρακτηριστικών - κατάστασης. Γενικά, σε μορφή λίστας μπορούν να αποθηκεύονται με επιτυχία αυτογραφικά όργανα, οπότε επιτυγχάνεται και η ζητούμενη οικονομία χώρου. Επίσης τα όργανα ραδιοβολίσεων αποθηκεύονται σε μορφή λίστας. Ένα όργανο μπορεί για ένα διάστημα να αποθηκεύεται σε κανονική μορφή και ένα άλλο σε μορφή λίστας. Η αλλαγή φαίνεται από τον πίνακα των χαρακτηριστικών των χρονοσειρών, απ' όπου ανακτάται η ζητούμενη μορφή αποθήκευσης (βλ. παρακάτω, πεδίο `format` πίνακα `timeseries_rpl`). Το να υποτίθεται πως ένα όργανο ή ακόμα ένας τύπος οργάνου αποθηκεύεται πάντα με μια σταθερή μορφή είναι υπεραπλούστευση που δεν αξιοποιεί τις δυνατότητες της ΒΔ.

Τα πεδία των τιμών μπορούν, και σε αυτή την περίπτωση, να αναλύονται περαιτέρω σε κ πεδία, ανάλογα με τον τρόπο μέτρησης του οργάνου. Δηλαδή ισχύει:

$$\langle \text{τιμή-}i \rangle = \langle \text{πεδίο-1} \rangle [\langle \text{πεδίο-2} \rangle \dots \langle \text{πεδίο-}k \rangle]$$

όπου τα πεδία είναι ακέραιοι διαφόρων ευρών (1, 2 ή 4 Bytes). Τελικά οι διάφορες μορφές λίστας, `lstxxx`, έχουν ως εξής:

```
instrument date status num interval value0 ... value31
      [ value32 ... valueλ ]
```

όπου $\lambda = 32 * k - 1$, δηλαδή για $k = 2$ (πχ. ανεμόμετρο), $\lambda = 63$.

Η μορφή λίστας δεν είναι δυσκολότερη στην επεξεργασία της από τις εφαρμογές από την κανονική μορφή. Μπορεί μάλιστα να είναι "διαφανής" { Γ } για τις εφαρμογές με την προσθήκη κατάλληλης ρουτίνας βιβλιοθήκης που μετατρέπει σε κανονική μορφή τη μορφή λίστας "ζετυλίγοντάς" την. Δηλαδή η εγγραφή:

```
instrument date status num interval value0 ... value31
```

θα διαβάζεται από τη ΒΔ, θα περνά ως παράμετρος στη ρουτίνα βιβλιοθήκης που θα επιστρέφει το σύνολο `num` εγγραφών σε "κανονική" μορφή:

```
instrument date status value0
instrument (date + 1 * interval) status value1
instrument (date + 2 * interval) status value2
...
```

instrument (date + (num-1) * interval) status value(num-1)

Θα πρέπει όμως η χρήση της μορφής λίστας να δικαιολογείται και από την επιτυγχανόμενη οικονομία χώρου με την αποθήκευση επαρκούς αριθμού τιμών. Πράγματι, η μορφή λίστας καταλαμβάνει σταθερό εύρος σε Bytes άσχετα με το πόσες τιμές πραγματικά αποθηκεύονται, μια και δεσμεύεται χώρος για 32 τιμές. Αν κάθε τιμή (με τα x πεδία της) έχει εύρος β Bytes, τότε το εύρος της μορφής λίστας είναι $4 + 4 + 4 + 1 + 4 + 32 * \beta$ Bytes (άθροιση των αντίστοιχων ευρών ακεραίων για όλα τα πεδία της μορφής). Αν οι num εγγραφές της χρονοσειράς αποθηκεύονταν σε κανονική μορφή θα καταλάμβαναν χώρο $(4 + 4 + 4 + \beta) * num$ Bytes. Θα πρέπει λοιπόν ο χώρος που καταλαμβάνει η εγγραφή μορφής λίστας να είναι τουλάχιστον μικρότερος από το χώρο των num ισοδύναμων εγγραφών κανονικής μορφής. Δηλαδή να ισχύει η ανισότητα:

$$num > (17 + 32 * \beta) / (12 + \beta)$$

Για παράδειγμα, για $\beta = 3$, δηλαδή 2 Bytes για την τιμή και 1 για πιθανό null (βλ. παρακάτω) θα πρέπει $num > 7.53$ δηλαδή θα πρέπει σε κάθε εγγραφή λίστας να αποθηκεύονται τουλάχιστον 8 τιμές για να συμφέρει από άποψη αποθηκευτικού χώρου και κυκλοφορίας στο δίκτυο η χρήση της.

3.7.6 Τιμές null. Όλες οι τιμές μετρήσεων που αποθηκεύονται στη ΒΔ μπορούν να πάρουν την ειδική τιμή null. Η null χαρακτηρίζεται ως "μη-τιμή" { Γ } που δηλώνει την έλλειψη - απουσία - απροσδιοριστία τιμής για το συγκεκριμένο πεδίο της ΒΔ. Σε συμβατικά συστήματα η έλλειψη τιμής υποδηλώνεται από κάποια ιδιαίτερη τιμή όπως -9999 κτλ. Στις σύγχρονες ΣΒΔ η σηματοδότηση της τιμής ενός πεδίου ως null έχει σαν πλεονέκτημα πως προσδιορίζει μονοσήμαντα την απουσία τιμής χωρίς ασάφειες και επίσης με άμεση υποστήριξη από τη γλώσσα SQL: τα πεδία null αγνοούνται στην περίπτωση εφαρμογής συναρτήσεων SQL όπως οι sum(), count(), avg() κοκ. Το μόνο μειονέκτημα της χρήσης των πεδίων null είναι η χρήση ενός επιπλέον Byte χώρου για κάθε "nullable" πεδίο, για να αποθηκεύεται η πληροφορία αν το πεδίο είναι null ή όχι.

Ενα σημαντικό χαρακτηριστικό της χρήσης null είναι το ακόλουθο: αν στη ΒΔ αποθηκεύεται μια χρονοσειρά που ορίζεται πως έχει κάποιο σταθερό χρονικό βήμα, τότε στην περίπτωση που μια τιμή - μέτρηση λείπει, μπορεί να υπάρχει στη ΒΔ το αντίστοιχο ζεύγος οργάνου - ημερομηνίας και η μέτρηση να παίρνει την τιμή null. Με αυτό τον τρόπο δεν διαταράσσεται η "συνέχεια" των χρονοσειρών. Εξάλλου, τα null είναι πολύ χρήσιμα στην περίπτωση γραμμικής αποθήκευσης (βλ. παρακάτω).

3.7.7 Γραμμική αποθήκευση. Πολλά από τα μεγέθη που αποθηκεύονται στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ παρουσιάζουν μια "γραμμική" συμπεριφορά. Αυτό σημαίνει το εξής: μετά

από κάποιο "σημείο καμπής" το μέγεθος αλλάζει την τιμή του με γραμμικό τρόπο ή και μένει σταθερό (υποπερίπτωση φυσικά της γραμμικής μεταβολής). Κλασσικό παράδειγμα τέτοιας συμπεριφοράς είναι η βροχή. Αλλά πάλι μεγέθη, όπως η θερμοκρασία, ακολουθούν τελείως τυχαία μεταβολή και δεν υπακούουν στο μοντέλο αυτό. Για τα μεγέθη όμως που υπακούουν προβλέπεται ο γραμμικός τρόπος αποθήκευσης. Ο γραμμικός τρόπος αποθήκευσης εφαρμόζεται μόνο στην κανονική μορφή αποθήκευσης και μόνο για χρονοσειρές με σταθερό βήμα - χρονική διακριτικότητα για μια σημαντική χρονική περίοδο. Η εφαρμογή του στην μορφή λίστας είναι δυνατή αλλά αυξάνει υπερβολικά την πολυπλοκότητα του συστήματος και την πιθανότητα λαθών. Ο γραμμικός τρόπος αποθήκευσης συνίσταται στην αποθήκευση μόνο των σημείων καμπής μιας γραμμικά μεταβαλλόμενης (με σταθερό βήμα) χρονοσειράς, δηλαδή μόνο των σημείων μετά τα οποία δεν ισχύει η γραμμική μεταβολή με ίδιες παραμέτρους ("κλίση" γραμμής). Όλα τα ενδιάμεσα σημεία είναι δυνατό να "αναδημιουργηθούν" στο επίπεδο της εφαρμογής με απλή γραμμική παρεμβολή μεταξύ των ακραίων σημείων για τις συγκεκριμένες ημερομηνίες για τις οποίες "αναμενόταν", λόγω του σταθερού βήματος της χρονοσειράς, να υπάρχει τιμή. Για να υπολογιστεί η τιμή για μια δεδομένη ημερομηνία αναζητείται το αποθηκευμένο στη ΒΔ ζεύγος (ημερομηνία, τιμή) για τη μέγιστη ημερομηνία μικρότερη ή ίση από τη ζητούμενη. Αν η ημερομηνία αυτής της τιμής είναι η ζητούμενη τότε βρέθηκε. Αλλιώς αναζητείται το αποθηκευμένο ζεύγος (ημερομηνία, τιμή) για την ελάχιστη ημερομηνία μεγαλύτερη ή ίση από τη ζητούμενη. Αν η ημερομηνία αυτή είναι η ζητούμενη, τότε βρέθηκε η τιμή. Αν όχι, τότε γίνεται γραμμική παρεμβολή ανάμεσα στις τιμές των δύο ημερομηνιών (σαν η χρονοσειρά να βρίσκεται πάνω σε ένα δισδιάστατο διάγραμμα, όπου στον άξονα των x αναπαριστάται ο χρόνος και στον άξονα των y αναπαριστάται η τιμή της χρονοσειράς) και βρίσκεται η ζητούμενη τιμή.

Τυχόν ελλείπουσες τιμές αποθηκεύονται ως null και αντιμετωπίζονται με αντίστοιχο τρόπο ως εξής: αποθηκεύεται το τελευταίο ζεύγος (ημερομηνία, τιμή) για το οποίο υπήρχε μέτρηση, στη συνέχεια το ζεύγος (ημερομηνία, null) για την πρώτη ημερομηνία για την οποία υπάρχει ελλείπουσα τιμή και έπειτα το ζεύγος (ημερομηνία, τιμή) για την πρώτη ημερομηνία που υπάρχει πάλι μέτρηση. Ανάμεσα στο δεύτερο και το τρίτο ζεύγος υποτίθεται έλλειψη τιμών. Και γενικά στο γραμμικό τρόπο αποθήκευσης υποτίθεται πως η χρονοσειρά εξελίσσεται γραμμικά στο χρόνο με κάποια σταθερή "κλίση" μέχρι να υπάρξει οποιαδήποτε αλλαγή στην "κλίση" ή να λείψει κάποια τιμή, οπότε το σημείο αυτό επίσης αποθηκεύεται.

Νέα "σημεία" - ζεύγη (ημερομηνία, τιμή) σε υπάρχουσα χρονοσειρά προστίθενται ως εξής: αν το νέο σημείο "ανήκει" στην ήδη υπάρχουσα "γραμμή", δηλαδή αν η τιμή του είναι ίση ή διαφορετική μέσα σε κάποια όρια ανοχής, πχ. 0.1% με την αναμενόμενη από τη γραμμική παρεμβολή (δηλαδή αν η αναμενόμενη τιμή πχ. για βροχή είναι 10, η υπάρχουσα τιμή να είναι από 9.99 ως 10.01) τότε δεν αλλάζει τίποτα: το ζεύγος θεωρείται πως έχει ήδη

αποθηκευθεί. Διαφορετικά, με βάση το γνωστό βήμα της χρονοσειράς βρίσκεται το αμέσως προηγούμενο και το αμέσως επόμενο ζεύγος της ήδη υπάρχουσας χρονοσειράς και αποθηκεύονται τόσο αυτά όσο και το νέο σημείο. Το αποτέλεσμα είναι πως, στη θέση του ενός αρχικού "ευθύγραμμου τμήματος" δημιουργούνται 4: το αρχικό διακόπτεται στο προηγούμενο ζεύγος και ξαναρχίζει στο επόμενο και δημιουργούνται και άλλα δύο, από το προηγούμενο ζεύγος μέχρι το νέο σημείο και από το νέο σημείο μέχρι το επόμενο ζεύγος.

Το αν μια χρονοσειρά αποθηκεύεται με το γραμμικό τρόπο φαίνεται από κατάλληλο πεδίο (linear) του πίνακα χαρακτηριστικών χρονοσειρών. Και σε αυτή την περίπτωση πρέπει να λαμβάνεται υπόψη πως μια χρονοσειρά ενός οργάνου ή όλες οι χρονοσειρές για τον ίδιο τύπο οργάνων δεν είναι ανάγκη να αποθηκεύονται ολόκληρες με τον ίδιο τρόπο, αλλά κατά περιόδους μπορεί να χρησιμοποιείται ή όχι η γραμμική αποθήκευση. Επιπλέον είναι δυνατή η δημιουργία ρουτίνας βιβλιοθήκης που με είσοδο τη γραμμική μορφή μιας χρονοσειράς θα επιστρέφει την πλήρη κανονική μορφή της, ώστε οι εφαρμογές να έχουν να κάνουν μόνο με την τελευταία.

3.7.8 Αποθήκευση δευτερογενών⁸. Η αποθήκευση των δευτερογενών γίνεται με τον ίδιο βασικό τρόπο με τα πρωτογενή. Χρησιμοποιούνται και σε αυτή την περίπτωση μορφές αποθήκευσης, αν και προτιμάται σαφώς η κανονική μορφή, με ή χωρίς γραμμική αποθήκευση. Σε περίπτωση που τα δεδομένα είναι μικρότερης διακριτότητας από ημερήσια (πχ. μηνιαία, ετήσια) πρέπει πάντα να χρησιμοποιείται η κανονική μορφή και μάλιστα χωρίς γραμμική αποθήκευση. Ενώ στην περίπτωση της αποθήκευσης πρωτογενών δεδομένων αυτά αποθηκεύονται με την ημερομηνία της μέτρησης, τα δευτερογενή αποθηκεύονται με μια συμβατική ημερομηνία που αναφέρεται στην περίοδο συνάδρσισης των δεδομένων, είτε στην αρχή είτε στο τέλος της. Για παράδειγμα, για μηνιαία / ετήσια διακριτότητα αποθηκεύεται ως ημερομηνία η πρώτη ημέρα του μήνα / έτους. Η ώρα - πρώτο λεπτό δεν έχει ιδιαίτερη σημασία, αρκεί να είναι πάντα η ίδια για το ίδιο όργανο και την ίδια διακριτότητα. Συνήθως χρησιμοποιείται μια συμβατική ώρα, ανάλογα με τον τύπο του οργάνου, πχ. 08:00 για τα βροχόμετρα, 14:00 για τα θερμομέτρα, 00:00 για άλλα όργανα. Η ερώτηση "ποιά είναι η τιμή του μεγέθους για τον m μήνα" ή "για το y έτος" εκφράζεται με χρήση της οριζόμενης από το χρήστη συνάρτησης SQL `hdate_part()`. Η αντίστοιχη ερώτηση SQL έχει τη μορφή:

```
select ... from ... where hdate_part('month', date) = m
```

ή

```
select ... from ... where hdate_part('year', date) = y
```

⁸ Στο κείμενο γενικά όπου χρησιμοποιούνται οι όροι "πρωτογενή", "δευτερογενή", "παράγωγα" εννοείται "... δεδομένα".

Για ημερήσια διακριτότητα ισχύουν τα ίδια ως προς την ώρα και τα πρώτα λεπτά: δεν έχουν σημασία αρκεί να παραμένουν "σταθερά".

Η λέξη κατάστασης των δευτερογενών δεδομένων έχει την ίδια ακριβώς μορφή με αυτήν των πρωτογενών και ο χειρισμός της γίνεται με τον ίδιο τρόπο. Αυτό βέβαια που κυρίως ενδιαφέρει στα δευτερογενή είναι το επίπεδο, καθώς οι έλεγχοι δεν έχουν ιδιαίτερο νόημα.

Όλοι οι πίνακες δευτερογενών αρχίζουν με το πρόθεμα `agg` που ενώνεται με το χαρακτήρα "_" με το όνομα της μορφής αποθήκευσης, κυρίως `stdxxx`, για να σχηματίσει το όνομα του πίνακα. Το γεγονός βέβαια πως στον ίδιο πίνακα δευτερογενών αποθηκεύονται χρονοσειρές με διαφορετικές διακριτότητες -ενώ στα πρωτογενή υπάρχει μια διακριτότητα, αυτή της μέτρησης- και πως για την ίδια ημερομηνία (πχ. πρώτη Ιανουαρίου) μπορούν να υπάρχουν διαφορετικά δεδομένα (πχ. μηνιαία για τον Ιανουάριο αλλά και ετήσια) καθιστά αναγκαία την εισαγωγή ενός ακόμα πεδίου που καθορίζει τη διακριτότητα της εγγραφής και την ταυτοποιεί. Το πεδίο αυτό είναι το `time_res` και περιέχει έναν ακέραιο ενός Byte που καθορίζει τη διακριτότητα σύμφωνα με τον παρακάτω πίνακα:

Τιμή <code>time_res</code>	Διακριτότητα (και όνομα πεδίου <code>time_res_t</code> στον πίνακα χαρακτηριστικών χρονοσειρών)
0	Ακανόνιστη (ΑΚΑΝΟΝ)
1	1 Λεπτό (1ΛΕΠΤΟ)
2	5 Λεπτά (5ΛΕΠΤΗ)
3	10 Λεπτά (10ΛΕΠΤΗ)
4	1 Ωρα (ΩΡΙΑΙΑ)
5	3 Ωρες (3ΩΡΗ)
6	8 Ωρες (8ΩΡΗ)
7	12 Ωρες (12ΩΡΗ)
8	Ημερήσια (ΗΜΕΡΗΣΙΑ)
9	Δεκαπενθήμερη (15ΘΗΜΕΡΗ)
10	Μηνιαία (ΜΗΝΙΑΙΑ)
11	Εξαμηνιαία (6ΜΗΝΙΑΙΑ)
12	Ετήσια (ΕΤΗΣΙΑ)
13	Διάφορα (ΔΙΑΦΟΡΑ)

14	30 Λεπτά (30ΛΕΠΤΗ)
15	6 Ωρες (6ΩΡΗ)

Πίνακας 3.7.8.1: Διαθέσιμες χρονικές διακριτότητες

Είναι προφανές πως στους πίνακες δευτερογενών μπορούν να αποθηκευθούν δεδομένα με διακριτότητα μόνο κάποια από αυτές που περιέχονται στον πίνακα. Οι συχνότερα χρησιμοποιούμενες διακριτότητες είναι οι 8 (ΗΜΕΡΗΣΙΑ), 10 (ΜΗΝΙΑΙΑ) και 12 (ΕΤΗΣΙΑ). Η ανάκτηση π.χ. μηνιαίων δευτερογενών γίνεται με εντολή SQL της μορφής:

```
select ... from ... where date ... and time_res = 10
```

Στην ίδια ημερομηνία μπορούν να υπάρχουν περισσότερες από μία τιμές, ανάλογα με τη διακριτότητα. Ενώ στα πρωτογενή καταβλήθηκε προσπάθεια να χρησιμοποιείται όπου είναι δυνατό η ίδια φυσική εγγραφή, στα δευτερογενή αυτό δεν είναι απαραίτητο, λόγω του σαφώς μικρότερου όγκου τους. Οπότε προτιμάται η απλούστερη λύση. Θα μπορούσε η διακριτότητα να γίνει μέρος της λέξης κατάστασης, κάτι τέτοιο όμως θα δημιουργούσε την ανάγκη διαφορετικών λέξεων κατάστασης πρωτογενών και δευτερογενών (αφού δεν υπάρχουν αρκετά ελεύθερα πεδία bits στη λέξη κατάστασης για αποθήκευση μέχρι 16 τιμών - δηλαδή 4 bits).

Τελικά οι πίνακες `aggr_stdxxx` (ή `aggr_1stxxx`) των δευτερογενών έχουν τη μορφή:

```
instrument date time_res status value0 [value1 ... valueλ]
```

όπου $\lambda = \kappa - 1$.

3.8 Αποθήκευση σταθερών

Η αποθήκευση σταθερών στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ δεν είναι δυνατό να έχει τη γενικότητα της αποθήκευσης των χρονοσειρών. Αυτό σημαίνει πως κάθε σταθερά έχει ιδιαιτερότητες που οδηγούν σε δικό της τρόπο αποθήκευσης. Ευτυχώς ο αριθμός των σταθερών είναι αρκετά μικρός ώστε να είναι δυνατή η αποθήκευση κάθε είδους σταθεράς σε ξεχωριστό πίνακα. Αυτό βέβαια σημαίνει πως η προσθήκη νέων ειδών σταθερών, σε αντίθεση με τυχόν νέα είδη οργάνων είναι επίπονη διαδικασία που πιθανότατα περιλαμβάνει και τη δημιουργία νέων πινάκων για την αποθήκευση των δεδομένων των σταθερών τόσο στις ΤΒΔ όσο και στις ισοδύναμες ΚΒΔ (νέες όψεις πινάκων τύπου `distr`).

Και στην περίπτωση των σταθερών υπάρχει η δυνατότητα τοπικής επεξεργασίας και αποθήκευσης σε πίνακες τύπου `x_`. Δηλαδή, όπως και στις χρονοσειρές, κάθε πίνακας

αποθήκευσης σταθερών έχει και έναν "τοπικό" πίνακα (που δεν συμμετέχει στην ΚΒΔ) με ίδιο βασικό όνομα και πρόθεμα x_. Ο πίνακας αυτός περιέχει στοιχεία σταθερών δεδομένων τα οποία μεν αναφέρονται σε αντικείμενα άλλων Φορέων αλλά έχουν υποστεί τοπική επεξεργασία. Τα σταθερά αυτά δεδομένα έχουν βέβαια και αντίστοιχες εγγραφές στους πίνακες - μητρώα χαρακτηριστικών σταθερών. Οι διάφορες σταθερές του ΥΔΡΟΣΚΟΠΙΟΥ φαίνονται στον πίνακα:

Σταθερά	Πλήρες όνομα σταθεράς	Πίνακας αποθήκευσης
ΛΙΘ-ΤΟΜ	ΛΙΘΟΛΟΓΙΚΗ ΤΟΜΗ	const litho sect
ΟΠΗ	ΟΠΗ	const drilling
ΣΩΛΗΝΩΣΗ	ΣΩΛΗΝΩΣΗ	const piping
ΔΙΑΤΟΜΗ	ΔΙΑΤΟΜΗ ΠΟΤΑΜΟΥ	const sect
ΥΔΡ-ΧΑΡ	ΥΔΡΑΥΛΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	const hydr
ΣΕΟ-ΤΑΜ	ΣΤΑΘΜΗ - ΕΠΙΦΑΝΕΙΑ - ΟΓΚΟΣ ΤΑΜΙΕΥΤ.	const lsv
ΣΠ-ΤΑΜ	ΣΤΑΘΜΗ - ΠΑΡΟΧΗ ΤΑΜΙΕΥΤΗΡΑ	const ld
ΠΑΡ-ΟΡΓ	ΠΑΡΑΓΩΓΟ ΟΡΓΑΝΟ	const dinstr

Πίνακας 3.8.1: Οι αποθηκευόμενες σταθερές

Στη συνέχεια αναλύεται ο τρόπος αποθήκευσης των σταθερών του ΥΔΡΟΣΚΟΠΙΟΥ.

3.8.1 Λιθολογική τομή. Η λιθολογική τομή ενός σταθμού $YYY \{ \Gamma \}$ είναι η περιγραφή της σύστασης του υπεδάφους κατά τη διαδρομή της γεώτρησης [Γεωργ93]. Η περιγραφή αυτή αποτελείται από το είδος και τα άλλα χαρακτηριστικά του πετρώματος, το βάθος στο οποίο αρχίζει το στρώμα του συγκεκριμένου πετρώματος και το βάθος στο οποίο τερματίζεται. Επειδή μάλιστα κάποιο στρώμα μπορεί να αποτελείται από συνδυασμό πετρωμάτων, υπάρχει και κάποιος αύξων αριθμός του πετρώματος. Τα πεδία του πίνακα `const_litho_sect` έχουν έτσι ως εξής:

3.8.1.1 Κωδικός σταθμού. Το πεδίο `obj0`, τύπου `hsiid`, είναι το αντικείμενο στο οποίο αναφέρεται κάθε σταθερά. Αν η σταθερά αναφέρεται σε σταθμό, το πεδίο αυτό περιλαμβάνει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του σταθμού. Αν αναφέρεται σε όργανο, τον κωδικό του οργάνου. Αν αναφέρεται σε χρονοσειρά (δεν υπάρχουν αυτή τη στιγμή τέτοιες σταθερές), τον κωδικό του οργάνου στο οποίο ανήκει η χρονοσειρά και σε πεδίο `obj1` υπάρχει ο α/α της χρονοσειράς. Στη συγκεκριμένη περίπτωση το πεδίο

obj0 περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του σταθμού στον οποίο αναφέρεται η σταθερά.

3.8.1.2 Αύξων αριθμός σταθεράς. Είναι ο αύξων αριθμός της σταθεράς για αυτό το αντικείμενο (obj0) και αυτή την υπηρεσία, όπως εμφανίζονται στον πίνακα των χαρακτηριστικών σταθερών. Ονομάζεται constant και είναι τύπου integer4.

3.8.1.3 Αύξων αριθμός στρώματος. Ο α/α του στρώματος πετρωμάτων. Δηλαδή το πρώτο στρώμα που βρέθηκε από το γεωτρήπανο (κοντούτερα στην επιφάνεια) είναι το υπ' αριθμό 1, το δεύτερο το 2 κ.κ. Κάθε τέτοιο στρώμα έχει τα χαρακτηριστικά που περιγράφονται από τα υπόλοιπα πεδία. Είναι τύπου integer1 (μέχρι 127 διαφορετικά στρώματα) και ονομάζεται seq (από το sequence).

3.8.1.4 Αρχή και τέλος στρώματος. Το βάθος σε m στο οποίο αρχίζει το συγκεκριμένο στρώμα πετρωμάτων και το βάθος στο οποίο τελειώνει είναι δύο πεδία, τα cfrom και cto, τύπου float4.

3.8.1.5 Αύξων αριθμός πετρώματος. Αν το στρώμα αποτελείται από περισσότερα του ενός πετρώματα στο πεδίο num_mat, τύπου integer1 υπάρχει ο α/α του συγκεκριμένου πετρώματος για το παρόν στρώμα πετρωμάτων. Το πρώτο -ή το μοναδικό, αν δεν υπάρχει συνδυασμός- πέτρωμα παίρνει τιμή 1, το δεύτερο 2 κ.κ. Προφανώς σε κάθε στρώμα μπορεί να υπάρχουν περισσότερα από ένα πετρώματα και άρα περισσότερες από μία εγγραφές στον πίνακα της λιθολογικής τομής.

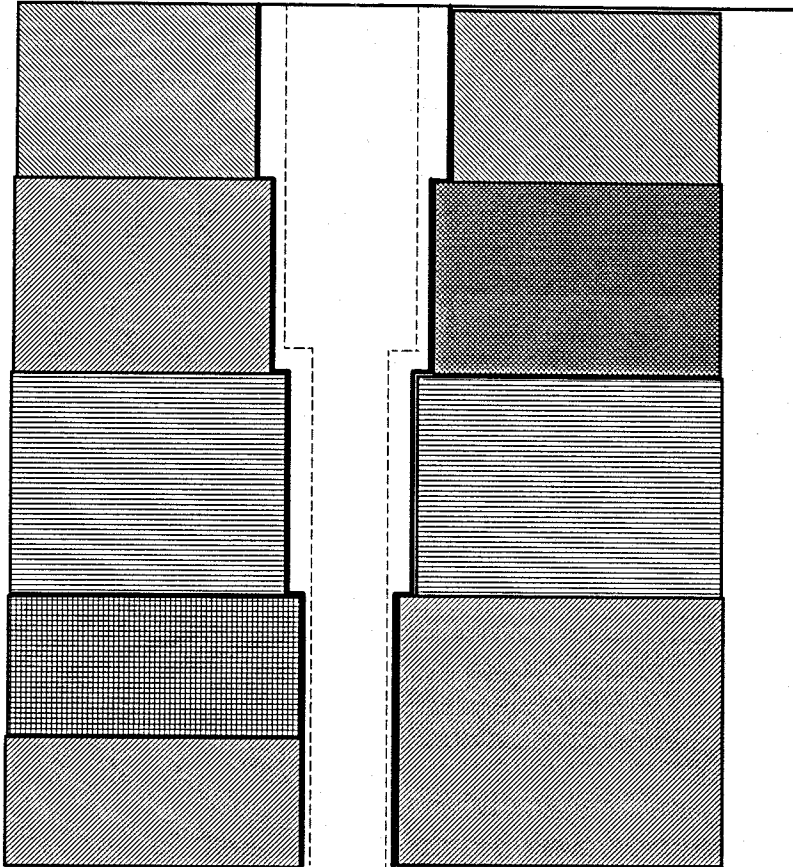
3.8.1.6 Είδος πετρώματος. Το είδος του πετρώματος της λιθολογικής τομής του σταθμού obj0 για α/α στρώματος seq, βάθος αρχής στρώματος cfrom, βάθος τέλους στρώματος cto και α/α πετρώματος στο στρώμα num_mat είναι litho_mat. Έχει τύπο char(8) και παίρνει τιμές από τον πίνακα αναφοράς - λεξικό όρων litho_mats (πεδίο litho_mat).

3.8.1.7 Χαρακτηριστικά πετρώματος. Τα χαρακτηριστικά του πετρώματος αυτού αποθηκεύονται σε 4 ακόμα πεδία char(8) (με αντίστοιχους πίνακες αναφοράς), τα litho_colour, litho_age, litho_size και litho_composition για το χρώμα, την ηλικία, το μέγεθος και τη σύσταση του πετρώματος.

3.8.2 Οπή. Η σταθερά αυτή αναφέρεται στη μορφή της οπής μιας γεώτρησης, δηλαδή σταθμού ΥΥΥ. Κάθε γεώτρηση αποτελείται από τμήματα, όπου κάθε τμήμα αρχίζει και τελειώνει σε κάποιο βάθος και έχει μια διάμετρο. Πιο αναλυτικά τα πεδία του πίνακα const_drilling είναι:

3.8.2.1 Κωδικός σταθμού. Πρόκειται για ένα πεδίο τύπου hsiid με ονομασία obj0 που περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του σταθμού ΥΥΥ στον οποίο αναφέρεται η σταθερά.

3.8.2.2 Αύξων αριθμός σταθεράς. Είναι ο αύξων αριθμός της σταθεράς για αυτό το αντικείμενο (obj0) και αυτή την υπηρεσία, όπως εμφανίζονται στον πίνακα των χαρακτηριστικών σταθερών. Ονομάζεται `constant` και είναι τύπου `integer4`.



Σχήμα 3.8.1: Οι διαφορετικές σταθερές YYY. Με χοντρή γραμμή η ΟΠΗ, με διακεκομμένη γραμμή η ΣΩΛΗΝΩΣΗ και με διαγραμμισμένα παραλληλόγραμμα τα πετρώματα της ΛΙΘ-TOM.

3.8.2.3 Αύξων αριθμός τμήματος. Είναι ο αύξων αριθμός του συγκεκριμένου τμήματος. Η τιμή 1 αντιστοιχεί στο τμήμα που αρχίζει στην επιφάνεια. Το πεδίο ονομάζεται `seq` και είναι τύπου `integer1` (μέχρι 127 τμήματα οπής).

3.8.2.4 Αρχή, τέλος και διάμετρος τμήματος. Το βάθος σε `m` στο οποίο αρχίζει το συγκεκριμένο τμήμα της οπής και το βάθος στο οποίο τελειώνει είναι δύο πεδία, τα `cfrom` και `cto`, τύπου `float4`. Επίσης το πεδίο `cdiameter`, τύπου `float4` είναι η διάμετρος του τμήματος σε `inches`.

3.8.3 Σωλήνωση. Η περιγραφή της σωλήνωσης μιας γεώτρησης σταδμού YYY είναι ο λόγος ύπαρξης αυτής της σταθεράς. Η σωλήνωση ακολουθεί γενικά τη μορφή της οπής

αλλά μπορεί να είναι και ανεξάρτητη από αυτήν ή και σε ορισμένα σημεία να μην υπάρχει καθόλου. Όπως και στην περίπτωση της οπής κάθε σωλήνωση χωρίζεται σε τμήματα, κάθε ένα από τα οποία έχει ορισμένα χαρακτηριστικά. Τα πεδία του πίνακα `const_piping` είναι:

3.8.3.1 Κωδικός σταθμού. Πρόκειται για ένα πεδίο τύπου `hsiid` με ονομασία `obj0` που περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του σταθμού `YYY` στον οποίο αναφέρεται η σταθερά.

3.8.3.2 Αύξων αριθμός σταθεράς. Είναι ο αύξων αριθμός της σταθεράς για αυτό το αντικείμενο (`obj0`) και αυτή την υπηρεσία, όπως εμφανίζονται στον πίνακα των χαρακτηριστικών σταθερών. Ονομάζεται `constant` και είναι τύπου `integer4`.

3.8.3.3 Αύξων αριθμός τμήματος. Είναι ο αύξων αριθμός του συγκεκριμένου τμήματος. Η τιμή 1 αντιστοιχεί στο τμήμα που αρχίζει στην επιφάνεια. Το πεδίο ονομάζεται `seq` και είναι τύπου `integer1` (μέχρι 127 τμήματα σωλήνωσης).

3.8.3.4 Αρχή, τέλος και διάμετρος τμήματος. Το βάθος σε `m` στο οποίο αρχίζει το συγκεκριμένο τμήμα της σωλήνωσης και το βάθος στο οποίο τελειώνει είναι δύο πεδία, τα `cfrom` και `cto`, τύπου `float4`. Επίσης το πεδίο `cdiameter`, τύπου `float4` είναι η διάμετρος του τμήματος σε `inches`.

3.8.3.5 Φίλτρο. Το αν το τμήμα αυτό της σωλήνωσης είναι φίλτρο φαίνεται από το πεδίο `cpipe_filter` που είναι τύπου `integer1` και παίρνει τιμές 1 (= 'TRUE') αν το τμήμα είναι φίλτρο και 0 (= 'FALSE') αν δεν είναι. Σε περίπτωση που είναι φίλτρο, το πεδίο `filter_t` τύπου `char(8)` με αντίστοιχο πίνακα αναφοράς τον `filter_ts` περιέχει τον τύπο του φίλτρου και το πεδίο `filter_gap` τύπου `float4` περιέχει το διάκενο του φίλτρου.

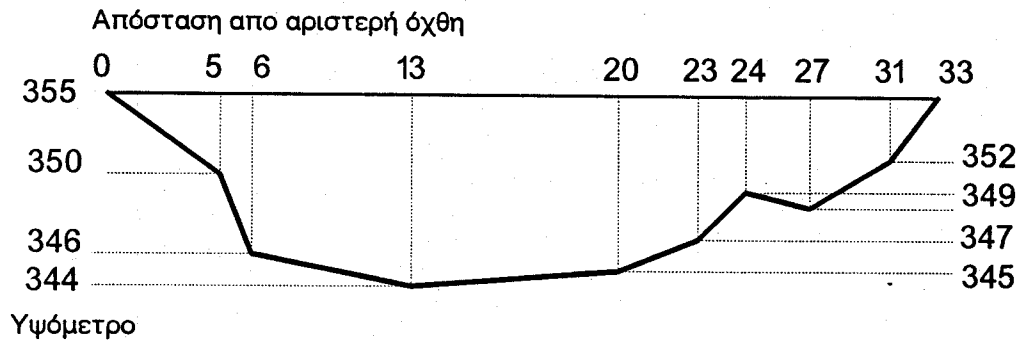
3.8.3.6 Υλικό. Το υλικό της σωλήνωσης στο συγκεκριμένο τμήμα περιέχεται στο πεδίο `cpipe_mat` τύπου `char(8)` με πίνακα αναφοράς τον `cpipe_mats`.

3.8.4 Διατομή. Κάθε υδρομέτρηση συνοδεύεται και από μια περιγραφή της διατομής του ποταμού τη στιγμή της υδρομέτρησης. Η διατομή αυτή αποθηκεύεται ως μια σειρά των σημείων που απαρτίζουν το βυθό του ποταμού. Οι συντεταγμένες των σημείων είναι η απόσταση από την αριστερή όχθη (όπως φαίνεται από τα κατόντη του ποταμού) και το απόλυτο υψόμετρο του σημείου ως προς τη στάθμη της θάλασσας. Αυτά αποθηκεύονται στον πίνακα `const_sect` ως ακολούθως:

3.8.4.1 Κωδικός οργάνου. Πρόκειται για ένα πεδίο τύπου `hsiid` με ονομασία `obj0` που περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του οργάνου υδρομέτρησης στο οποίο αναφέρεται η σταθερά. Κάθε υδρομέτρηση κανονικά συνοδεύεται από μια σταθερά διατομής, ωστόσο πολλές φορές η διατομή του ποταμού παραμένει σταθερή

(πχ. διότι ο ποταμός είναι εγκλιβωτισμένος) και άρα χρησιμοποιούνται διατομές προηγούμενων υδρομετρήσεων.

3.8.4.2 Αύξων αριθμός σταθεράς. Είναι ο αύξων αριθμός της σταθεράς για αυτό το αντικείμενο (`obj0`) και αυτή την υπηρεσία, όπως εμφανίζονται στον πίνακα των χαρακτηριστικών σταθερών. Ονομάζεται `constant` και είναι τύπου `integer4`.



Σχήμα 3.8.4.1: Διατομή ποταμού ως σύνολο σημείων με συντεταγμένες (απόσταση, υψόμετρο).

3.8.4.3 Αύξων αριθμός σημείου. Είναι ο αύξων αριθμός του συγκεκριμένου σημείου. Το σημείο 1 είναι αυτό που αντιστοιχεί στην αριστερή όχθη, έχει δηλαδή απόσταση 0. Το πεδίο ονομάζεται `seq` και είναι τύπου `integer4`.

3.8.4.4 Απόσταση και υψόμετρο. Πρόκειται για τις συντεταγμένες του σημείου. Είναι δύο πεδία `float4`, τα `distance` και `altitude`, που περιέχουν την απόσταση από την αριστερή όχθη και το υψόμετρο του συγκεκριμένου σημείου `seq`.

3.8.5 Υδραυλικά χαρακτηριστικά. Άλλη μια σταθερά που συνδέεται με τις υδρομετρήσεις, τα υδραυλικά χαρακτηριστικά του ποταμού δίνουν μια εικόνα της κοίτης του καθώς και γενικά στοιχεία για τη διατομή του. Τα υδραυλικά χαρακτηριστικά αποθηκεύονται για να δίνεται η δυνατότητα εναλλακτικού υπολογισμού της παροχής από τις στάθμες με υδραυλικές μεθόδους. Ο πίνακας `const_hydr` έχει τα ακόλουθα πεδία:

3.8.4.1 Κωδικός οργάνου. Πρόκειται για ένα πεδίο τύπου `hsiid` με ονομασία `obj0` που περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του οργάνου υδρομέτρησης στο οποίο αναφέρεται η σταθερά. Ισχύουν τα ίδια με την διατομή για τη σχέση των σταθερών με τις μετρήσεις.

3.8.4.2 Αύξων αριθμός σταθεράς. Είναι ο αύξων αριθμός της σταθεράς για αυτό το αντικείμενο (`obj0`) και αυτή την υπηρεσία, όπως εμφανίζονται στον πίνακα των χαρακτηριστικών σταθερών. Ονομάζεται `constant` και είναι τύπου `integer4`.

3.8.4.3 Κλίση. Η κλίση του ποταμού αποθηκεύεται στο πεδίο `slope` με τύπο `float4`.

3.8.4.4 Συντελεστής Manning. Ο συντελεστής Manning της κοίτης του ποταμού στο σημείο της υδρομέτρησης αποθηκεύεται στο πεδίο `manning` τύπου `float4`.

3.8.4.5 Στοιχεία διατομής. Το συνολικό πλάτος της διατομής του ποταμού αποθηκεύεται στο `float4` πεδίο `width`. Το ολικό εμβαδόν της διατομής αποθηκεύεται στο πεδίο `area` με τύπο `float4`. Τέλος, το μέσο βάθος της διατομής είναι το πεδίο `depth` τύπου `float4`. Αν η διατομή ήταν παραλληλόγραμμη θα ίσχυε $area = width * depth$.

3.8.6 Στοιχεία στάθμης - επιφάνειας - όγκου ταμιευτήρα. Οι καμπύλες στάθμης - επιφάνειας - όγκου των ταμιευτήρων είναι σταθερές που αναφέρονται σε κάποιο σταθμό που βρίσκεται στην περιοχή του ταμιευτήρα. Στην πραγματικότητα αποθηκεύονται ως τριάδες τιμών στον πίνακα `const_1sv` ως εξής:

3.8.6.1 Κωδικός σταθμού. Πρόκειται για ένα πεδίο τύπου `hsiid` με ονομασία `obj0` που περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του σταθμού του ταμιευτήρα στον οποίο αναφέρεται η σταθερά.

3.8.6.2 Αύζων αριθμός σταθεράς. Είναι ο αύζων αριθμός της σταθεράς για αυτό το αντικείμενο (`obj0`) και αυτή την υπηρεσία, όπως εμφανίζονται στον πίνακα των χαρακτηριστικών σταθερών. Ονομάζεται `constant` και είναι τύπου `integer4`.

3.8.6.3 Αύζων αριθμός τριάδας. Είναι ο αύζων αριθμός της συγκεκριμένης τριάδας. Το πεδίο ονομάζεται `seq` και είναι τύπου `integer4`.

3.8.6.4 Τριάδα στάθμης - επιφάνειας - όγκου. Σε τρία πεδία αποθηκεύεται μια τριάδα τιμών για τα ζητούμενα φυσικά μεγέθη. Στο `integer4` πεδίο `level` αποθηκεύεται η στάθμη του ταμιευτήρα σε `cm`. Στο `float4` πεδίο `surface` αποθηκεύεται η επιφάνεια του ταμιευτήρα σε `km2` και στο `integer4` πεδίο `volume` ο όγκος του νερού του ταμιευτήρα σε `m3` για αυτό το ζεύγος στάθμης - επιφάνειας.

3.8.7 Καμπύλη στάθμης - παροχής υπερχειλιστή ταμιευτήρα. Η καμπύλη στάθμης - παροχής υπερχειλιστή του ταμιευτήρα αποθηκεύεται ως μια σειρά σημείων στάθμης - παροχής που αναφέρονται σε έναν σταθμό του ταμιευτήρα, όπως και για τα στοιχεία στάθμης - επιφάνειας - όγκου. Πρόκειται για "κατασκευαστική" σταθερά του συγκεκριμένου φράγματος. Αποθηκεύεται στον πίνακα `const_1d` ως εξής:

3.8.7.1 Κωδικός σταθμού. Πρόκειται για ένα πεδίο τύπου `hsiid` με ονομασία `obj0` που περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του σταθμού του ταμιευτήρα στον οποίο αναφέρεται η σταθερά.

3.8.7.2 Αύξων αριθμός σταθεράς. Είναι ο αύξων αριθμός της σταθεράς για αυτό το αντικείμενο (`obj0`) και αυτή την υπηρεσία, όπως εμφανίζονται στον πίνακα των χαρακτηριστικών σταθερών. Ονομάζεται `constant` και είναι τύπου `integer4`.

3.8.7.3 Αύξων αριθμός σημείου. Είναι ο αύξων αριθμός του συγκεκριμένου σημείου της καμπύλης στάθμης - παροχής υπερχειλιστή. Το πεδίο ονομάζεται `seq` και είναι τύπου `integer4`.

3.8.7.4 Στάθμη και παροχή. Στα `integer4` πεδία `level` και `discharge` αποθηκεύονται αντίστοιχα η στάθμη σε `cm` και η παροχή υπερχειλιστή για αυτή τη στάθμη σε `m3/s`.

3.8.8 Παράγωγο όργανο. Οι "μετρήσεις" κάθε παράγωγου οργάνου είναι το αποτέλεσμα συγκεκριμένων πράξεων πάνω σε μετρήσεις άλλων πραγματικών ή ακόμα και παράγωγων οργάνων. Τα όργανα που "αποτελούν" το παράγωγο όργανο είναι πάντα τα ίδια και περιγράφονται ως σταθερά που αντιστοιχεί στο συγκεκριμένο παράγωγο όργανο. Αποθηκεύονται στον πίνακα `const_dinstr` με πεδία:

3.8.8.1 Κωδικός οργάνου. Πρόκειται για ένα πεδίο τύπου `hsiid` με ονομασία `obj0` που περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του παράγωγου οργάνου

3.8.8.2 Αύξων αριθμός σταθεράς. Είναι ο αύξων αριθμός της σταθεράς για αυτό το αντικείμενο (`obj0`) και αυτή την υπηρεσία, όπως εμφανίζονται στον πίνακα των χαρακτηριστικών σταθερών. Ονομάζεται `constant` και είναι τύπου `integer4`.

3.8.8.3 Αύξων αριθμός σημείου. Είναι ο αύξων αριθμός του συγκεκριμένου "βασικού" οργάνου. Το πεδίο ονομάζεται `seq` και είναι τύπου `integer4`.

3.8.8.4 Βασικό όργανο. Σε αυτό το τύπου `hsiid` πεδίο με όνομα `base_instrument` αποθηκεύεται ο κωδικός ΥΔΡΟΣΚΟΠΙΟΥ του υπ' αριθμό `seq` βασικού οργάνου που απαρτίζει, μαζί με τα άλλα όργανα που περιγράφονται στη σταθερά, το παράγωγο όργανο. Ο τρόπος με τον οποίο γίνεται η παραγωγή δεν περιγράφεται, καθώς θεωρείται πως είναι γνωστός και σταθερός για κάθε είδος παράγωγου οργάνου (πχ. εξατμισοδιαπνοή, παροχή).

3.9 Είδη μεταβλητών και οργάνων

Η ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ μπορεί να αποθηκεύσει ένα μεγάλο αριθμό διαφορετικών φυσικών μεταβλητών οι οποίες μετρώνται άμεσα ή έμμεσα (από παρατήρηση ή υπολογισμό) από διάφορα όργανα. Σε κάθε περίπτωση, ακόμα και αν υπάρχουν μεταβλητές ή όργανα που δεν έχουν περιληφθεί, είναι πολύ εύκολο να περιληφθούν με απλή εισαγωγή τους (μέσω πχ. εντολών SQL `insert`) στους αντίστοιχους πίνακες (`variables` για τις μεταβλητές, `instrument_ts` και `precision_unit` για τα όργανα).

3.9.1 Μεταβλητές. Οι αποθηκευόμενες στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ μεταβλητές είναι προς το παρόν 60 και φαίνονται στον πίνακα variables:

Μεταβλητή	Πλήρες όνομα μεταβλητής
Δ-ΑΝΘΡ	ΔΙΟΞΕΙΔΙΟ ΑΝΘΡΑΚΑ
ΥΔΡΟΘ	ΥΔΡΟΘΕΙΟ
ΡΗ	ΡΗ
Δ-ΠΥΡΙΤ	ΔΙΟΞΕΙΔΙΟ ΠΥΡΙΤΙΟΥ
ΑΓΩΓΙΜΟΤ	ΑΓΩΓΙΜΟΤΗΤΑ
ΑΚΤΙΝΟΒ	ΑΚΤΙΝΟΒΟΛΙΑ
ΑΜΜΩΝΙΑ	ΑΜΜΩΝΙΑ
ΑΝΕΜΟΣ	ΑΝΕΜΟΣ
ΑΝΘΡΑΚΚΑ	ΑΝΘΡΑΚΙΚΑ
ΑΡΓΙΛΙΟ	ΑΡΓΙΛΙΟ
ΑΡΓΙΛΟΣ	ΑΡΓΙΛΟΣ
ΑΡΣΕΝΙΚΟ	ΑΡΣΕΝΙΚΟ
ΑΣΒΕΣΤΙΟ	ΑΣΒΕΣΤΙΟ
ΒΡΟΧΗ	ΒΡΟΧΗ
ΔΟΚ-ΑΝΤΛ	ΔΟΚΙΜΑΣΤΙΚΕΣ ΑΝΤΛΗΣΕΙΣ
ΔΡΟΣΟΣ	ΔΡΟΣΟΣ
ΔΥΝΟΞΕΙΑ	ΔΥΝΑΜΙΚΟ ΟΞΕΙΔΟΑΝΑΓΩΓΗΣ
ΕΞΑΤΜΙΣΗ	ΕΞΑΤΜΙΣΗ
ΗΛΙΟΦΑΝ	ΗΛΙΟΦΑΝΕΙΑ
ΘΕΙΙΚΑ	ΘΕΙΙΚΑ
ΘΕΡΜΟΚ	ΘΕΡΜΟΚΡΑΣΙΑ
ΘΕΡΜΟΚ-Υ	ΘΕΡΜΟΚΡΑΣΙΑ ΥΓΡΟΥ ΑΕΡΑ
ΘΕΡΜΟΚΡΕ	ΘΕΡΜΟΚΡΑΣΙΑ ΥΥΥ

ΘΡΜΕΔΑΦ	ΘΕΡΜΟΚΡΑΣΙΑ ΕΔΑΦΟΥΣ
ΙΧΝ-ΥΕΤ	ΙΧΝΗ ΥΕΤΟΥ
ΚΑΔΜΙΟ	ΚΑΔΜΙΟ
ΚΑΙΡΟΣ	ΚΑΙΡΟΣ
ΚΑΛΙΟ	ΚΑΛΙΟ
ΚΑΤ-ΕΔΑΦ	ΚΑΤΑΣΤΑΣΗ ΕΔΑΦΟΥΣ
ΜΑΓΓΑΝΙΟ	ΜΑΓΓΑΝΙΟ
ΜΑΓΝΗΣΙΟ	ΜΑΓΝΗΣΙΟ
ΜΟΛΥΒΔΟΣ	ΜΟΛΥΒΔΟΣ
ΝΑΤΡΙΟ	ΝΑΤΡΙΟ
ΝΕΦΩΣΗ	ΝΕΦΩΣΗ
ΝΙΚΕΛΙΟ	ΝΙΚΕΛΙΟ
ΝΙΤΡΙΚΑ	ΝΙΤΡΙΚΑ
ΝΙΤΡΩΔΗ	ΝΙΤΡΩΔΗ
ΕΗΡΟΥΠΟΛ	ΕΗΡΟ ΥΠΟΛΕΙΜΜΑ
ΟΞ-ΑΝΘΡΚ	ΟΞΙΝΑ ΑΝΘΡΑΚΙΚΑ
ΟΡΑΤΟΤΗΣ	ΟΡΑΤΟΤΗΤΑ
ΠΑΡΟΧΗ	ΠΑΡΟΧΗ
ΠΙΕΣΗ	ΑΤΜΟΣΦΑΙΡΙΚΗ ΠΙΕΣΗ
ΣΙΔΗΡΟΣ	ΣΙΔΗΡΟΣ
ΣΚΛΗΡΤΗΤ	ΣΚΛΗΡΟΤΗΤΑ
ΣΤΑΘΜΗ	ΣΤΑΘΜΗ
ΣΤΘ-ΣΗΜ	ΣΤΑΘΜΗ ΣΗΜΕΙΟΥ ΥΔΑΤΟΣ ΥΥΥ
ΣΤΠΑΡΟΧΗ	ΣΤΕΡΕΟΠΑΡΟΧΗ
ΣΧΕΤ-ΥΓΡ	ΣΧΕΤΙΚΗ ΥΓΡΑΣΙΑ
ΥΔΡΑΡΓΥΡ	ΥΔΡΑΡΓΥΡΟΣ
ΥΨΟΜΕΤΡΟ	ΥΨΟΜΕΤΡΟ
ΦΘΟΡΙΟ	ΦΘΟΡΙΟ

ΦΩΣΦΟΡΚΑ	ΦΩΣΦΟΡΙΚΑ
ΧΑΛΚΟΣ	ΧΑΛΚΟΣ
ΧΙΟΝΙ	ΧΙΟΝΙ
ΧΛΩΡΙΟ	ΧΛΩΡΙΟ
ΧΡΩΜΙΟ	ΧΡΩΜΙΟ
ΨΕΥΔΑΡΓΡ	ΨΕΥΔΑΡΓΥΡΟΣ
ΤΑΣΥΔΡ	ΤΑΣΗ ΥΔΡΑΤΜΩΝ
ΥΓΡΑΣΙΑ	ΥΓΡΑΣΙΑ
ΤΡΟΠΟΠ	ΤΡΟΠΟΠΑΥΣΗ

Πίνακας 3.9.1.1: Οι αποθηκευόμενες μεταβλητές - φυσικά μεγέθη

3.9.2 Όργανα. Τα όργανα που μετρούν τις διάφορες μεταβλητές περιέχονται στον πίνακα `instrument_ts` (όπως σε όλη τη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ το επίθεμα `{ Γ } _t` προέρχεται από τον αγγλικό όρο "type" δηλαδή τύπος και το `_ts` είναι, όπως ήδη αναφέρθηκε, πληθυντικός, "types". Δηλαδή υπάρχει ένας πίνακας `_ts` του οποίου το βασικό πεδίο είναι `_t`). Στον πίνακα `precision_unit` περιέχονται επιπλέον χαρακτηριστικά για κάθε όργανα και συγκεκριμένα η απαιτούμενη ακρίβεια και η μονάδα μέτρησης για κάθε πεδίο της μετρούμενης τιμής, το αν το όργανο μετρά απευθείας ή από ανθρώπινη παρατήρηση ή είναι παράγωγο καθώς και οι δυνατές μορφές αποθήκευσης για αυτό το όργανο.

Στη συνέχεια αναλύεται ο τρόπος που αποθηκεύονται οι χρονοσειρές των διαφόρων οργάνων, ανάλογα με το είδος της μετρούμενης μεταβλητής.

3.9.2.1 Στάθμη. Πρόκειται για δεδομένα που αναφέρονται σε μετρήσεις στάθμης, δηλαδή της μεταβλητής ΣΤΑΘΜΗ αλλά και της μεταβλητής ΣΤΘ-ΣΗΜ της ΥΥΥ. Τα όργανα που μετρούν (απευθείας) στάθμη την αποθηκεύουν με μορφές αποθήκευσης ενός πεδίου, στο οποίο και αποθηκεύουν τη μετρηθείσα τιμή της στάθμης. Είναι τα ακόλουθα:

- **ΣΤΘΜΤΡ, ΣΤΘΓΡ.** Το ΣΤΑΘΜΗΜΕΤΡΟ και ο ΣΤΑΘΜΗΓΡΑΦΟΣ αποθηκεύουν σε μορφές `std4` και `1st4` σε μονάδες `mm` με ακρίβεια 0 δεκαδικών.
- **ΣΤΘ-ΣΗΜ.** Το όργανο ΣΤΑΘΜΗ ΣΗΜΕΙΟΥ ΥΔΑΤΟΣ ΥΥΥ μετρά τη στάθμη σταθμών ΥΥΥ, δηλαδή πηγών και γεωτρήσεων. Η μορφή αποθήκευσης είναι `std4` ή `1st4`, η ακρίβεια είναι 2 δεκαδικά και η μονάδα το `m`.

3.9.2.2 Καιρός. Τα δεδομένα καιρού αναφέρονται στη μεταβλητή ΚΑΙΡΟΣ και προκύπτουν από παρατηρήσεις του καιρού -όχι από απευθείας μετρήσεις. Αναφέρονται γενικά σε δύο στοιχεία, έναν παρόντα και έναν παρελθόντα καιρό. Υπάρχουν δύο όργανα καιρού:

- **ΚΑΙΡΟΣ.** Το όργανο ΚΑΙΡΟΣ αποθηκεύει σε μορφή `std11` ή `1st11` δεδομένα καιρού. Στο πρώτο πεδίο αποθηκεύεται ο παρών και στο δεύτερο ο παρελθών καιρός. Οι τιμές που αποθηκεύονται είναι μικροί ακέραιοι (1 Byte) που αντιστοιχούν στον κώδικα WMO 4677. Για την ερμηνεία τους και τη μετατροπή τους σε "αναγνώσιμη" μορφή, ακολουθείται σε αυτήν και σε όλες τις περιπτώσεις αποθήκευσης κωδικών αντί "κανονικών" τιμών η ακόλουθη μεθοδολογία: στον πίνακα `codes` αποθηκεύονται όλοι οι μικροί αυτοί ακέραιοι - κώδικες στο πεδίο `code`. Στο πεδίο `code_t` για κάθε εγγραφή - κώδικα αποθηκεύεται το είδος του κώδικα (πχ. ΚΑΙΡΟΣ, ΝΕΦΩΣΗ, ΠΙΕΣΗ, ΑΝΤΛΙΑ) και στο πεδίο `descr` η ερμηνεία του κώδικα που περιέχεται στο πεδίο `code` για το είδος `code_t`. Η εφαρμογή που θέλει να παρουσιάσει στο χρήστη την ερμηνεία του κώδικα αρκεί να ανακτήσει από τον πίνακα `codes` το πεδίο `descr` για το είδος κώδικα `code_t`. Πχ. για τον καιρό του οποίου το είδος κώδικα είναι ΚΑΙΡΟΣ θα είναι μια εντολή SQL της μορφής:

```
select ..., c.descr from raw_std11 r, codes c where ... and
c.code = r.value0 and c.code_t = 'ΚΑΙΡΟΣ'
```

Όπως είναι φυσικό, δεν υπάρχει μονάδα μέτρησης και η ακρίβεια είναι 0.

- **ΚΑΙΡΟΣ-Σ.** Ο ΚΑΙΡΟΣ - ΣΥΜΒΟΛΑ είναι ένα όργανο που καταγράφει, με τη μορφή συμβόλων καιρού, τις παρατηρήσεις τυχόν φαινομένων που σχετίζονται με τον παρόντα και τον παρελθόντα καιρό. Τα σύμβολα είναι και αυτά κώδικας της EMY [EMY80]. Ο τύπος του κώδικα είναι ΚΑΙΡΟΣ-Σ. Επειδή μέχρι 3 σύμβολα μπορεί να σχετίζονται με τον παρόντα και 3 με τον παρελθόντα καιρό, η αποθήκευση γίνεται σε μορφή `std6` ή `1st6` με ακρίβεια 0 και βέβαια χωρίς μονάδες μέτρησης. Αυτό που πρέπει να συμβαίνει για κάθε εγγραφή αυτού του οργάνου είναι να υπάρχει μια εγγραφή ενός οργάνου τύπου ΚΑΙΡΟΣ του ίδιου σταθμού για την ίδια ημερομηνία, στην οποία εγγραφή να αντιστοιχίζονται τα σύμβολα καιρού, τα 3 πρώτα στον παρόντα καιρό και τα άλλα 3 στον παρελθόντα. Για παράδειγμα να είναι:

Εγγραφή οργάνου ΚΑΙΡΟΣ-Σ:

254, 5/12/1994 06:00, ..., 2, 1, null, 8, null, null⁹

⁹ Στις περιπτώσεις που δίνονται παραδείγματα εγγραφών, τα πεδία χωρίζονται μεταξύ τους με ';' (κόμμα). Δηλαδή στην περίπτωση αυτή το πεδίο `instrument` παίρνει τιμή 254, το πεδίο `date`

Απαραίτητη "συνδυασμένη" εγγραφή οργάνου ΚΑΙΡΟΣ του ίδιου σταθμού:

252, 5/12/1994 06:00, ..., 61, 75

Αυτό σημαίνει πως για τη συγκεκριμένη ημερομηνία, στον κώδικα 60 (σημαίνει 'Βροχή ασθενής συνεχής') του παρόντος καιρού αντιστοιχούν τα σύμβολα καιρού 2 (σημαίνει 'Βροχή') και 1 (σημαίνει 'Ψεκάδες') και δεν υπάρχει τρίτο διαδέσιμο σύμβολο ενώ στον κώδικα 75 (σημαίνει 'Ισχυρή συνεχής πτώση υιφάδων χιονιού') του παρελθόντος καιρού αντιστοιχεί το σύμβολο καιρού με κώδικα 8 (σημαίνει "Κοκκώδες χιόνι") και κανένα άλλο σύμβολο.

Πρακτικά πρέπει κάθε όργανο τύπου ΚΑΙΡΟΣ να συνοδεύεται κατά τη δημιουργία του και από ένα όργανο τύπου ΚΑΙΡΟΣ-Σ.

3.9.2.3 Βροχή και χιόνι. Η μεταβλητή ΒΡΟΧΗ αποθηκεύεται γενικά σε ένα πεδίο που περιέχει την τιμή που μετρήθηκε απευθείας από το όργανο. Το ίδιο ισχύει και για το ΧΙΟΝΙ. Τα όργανα που μετρούν είναι:

- ΒΡΧΜΤΡ, ΒΡΧΓΡ, ΑΘΒΡΧΜΤΡ. Το ΒΡΟΧΟΜΕΤΡΟ, ο ΒΡΟΧΟΓΡΑΦΟΣ και το ΑΘΡΟΙΣΤΙΚΟ ΒΡΟΧΟΜΕΤΡΟ αποθηκεύονται σε μορφή std2 ή 1st2 με ακρίβεια 1 δεκαδικό και σε μονάδες mm.
- ΔΥΕΤΟΥ. Η ΔΙΑΡΚΕΙΑ ΥΕΤΟΥ είναι ένα όργανο που συνδέεται με αντίστοιχο όργανο που μετρά βροχή ή χιόνι. Δηλαδή, όπως πχ. με τα σύμβολα καιρού, κάθε εγγραφή ΔΥΕΤΟΥ αντιστοιχίζεται σε μια εγγραφή βροχόμετρου, χιονόμετρου κτλ του ίδιου σταθμού και για την ίδια ημερομηνία. Η τιμή της εγγραφής δίνει τη διάρκεια σε πρώτα λεπτά του υετού που έδωσε το ύψος υετού που υπάρχει στη συνδυασμένη εγγραφή, ώστε να μπορεί να υπολογιστεί η ένταση. Για παράδειγμα η εγγραφή ΒΡΧΜΤΡ:

256, 5/12/1994 08:00, ..., 154

μπορεί να συνοδεύεται από εγγραφή ΔΥΕΤΟΥ:

259, 5/12/1994 08:00, ..., 180

που σημαίνει πως η βροχή 15.4 (= 154 / 10¹) mm που μετρήθηκε τη συγκεκριμένη ημερομηνία προήλθε από υετό διάρκειας 180 πρώτων λεπτών. Κατά συνέπεια το όργανο αυτό δεν δημιουργεί χρονοσειρές και μετρά ακανόνιστα, με ακρίβεια 0 και μονάδα μέτρησης το min.

- ΙΧΝ-ΥΕΤ. Το όργανο ΙΧΝΗ ΥΕΤΟΥ αποθηκεύει σε μορφή std1 ή 1st1 χωρίς μονάδες και με ακρίβεια 0 δεκαδικών. Πρακτικά οποιαδήποτε εγγραφή αυτού του

παίρνει τιμή 5/12/1994 06:00, το πεδίο status αδιάφορη τιμή (...), το πεδίο value0 τιμή 2, το value1 τιμή 1, το value2 null κοκ.

οργάνου για κάποια ημερομηνία με το πρώτο πεδίο (value0) να έχει οποιαδήποτε τιμή διάφορη του 0 σημαίνει ύπαρξη (μη μετρήσιμων από τα άλλα όργανα - βροχόμετρα, χιονόμετρα κοκ-) ιχνών υετού σε αυτή την ημερομηνία. Είναι σαφές πως οι μετρήσεις αυτού του οργάνου είναι ακανόνιστες και δεν μπορούν να σχηματίσουν χρονοσειρά.

- **ΧΝΜΤΡ, ΧΝΤΡΠ, ΧΝΓΡ, ΧΝΠΛΚ, ΣΧΝΤΡΠ.** Τα όργανα ΧΙΟΝΟΜΕΤΡΟ, ΧΙΟΝΟΤΡΑΠΕΖΑ, ΧΙΟΝΟΓΡΑΦΟΣ, ΧΙΟΝΟΠΛΑΚΑ και ΣΥΣΣΩΡΕΥΤΙΚΗ ΧΙΟΝΟΤΡΑΠΕΖΑ μετρούν ΧΙΟΝΙ με ακρίβεια 1 δεκαδικού σε cm και αποθηκεύουν σε μορφή std2 ή lst2.
- **ΧΝΒΡΧΜΤΡ, ΧΝΒΡΧΓΡ.** Τα όργανα ΧΙΟΝΟΒΡΟΧΟΜΕΤΡΟ και ΧΙΟΝΟΒΡΟΧΟΓΡΑΦΟΣ μετρούν ΧΙΟΝΙ ανηγμένη σε ισοδύναμα mm βροχής με ακρίβεια 1 δεκαδικό και αποθηκεύουν σε μορφή std2 ή lst2.

3.9.2.4 Θερμοκρασία ξηρού και υγρού αέρα. Τα όργανα που μετρούν ΘΕΡΜΟΚ και ΘΕΡΜΟΚ-Υ αποθηκεύουν γενικά σε ένα πεδίο την απευθείας μετρηθείσα τιμή. Η ακρίβεια είναι 1 δεκαδικό γρηφίο και η μονάδα μέτρησης ο βαθμός C. Οι μορφές αποθήκευσης είναι γενικά οι std2 και lst2.

- **ΘΡΜΜΤΡ, ΘΡΜΓΡ.** Όργανα ΘΕΡΜΟΜΕΤΡΟ και ΘΕΡΜΟΓΡΑΦΟΣ
- **ΘΡΜΜΤΡΜ, ΘΡΜΜΤΡΕ.** Όργανα ΘΕΡΜΟΜΕΤΡΟ ΜΕΓΙΣΤΩΝ και ΘΕΡΜΟΜΕΤΡΟ ΕΛΑΧΙΣΤΩΝ. Μετρούν και αποθηκεύουν αντίστοιχα τη μέγιστη και την ελάχιστη θερμοκρασία ξηρού αέρα.
- **ΘΡΜΜΤΡΜΕ.** Είναι το ΘΕΡΜΟΜΕΤΡΟ ΜΕΓΙΣΤΩΝ - ΕΛΑΧΙΣΤΩΝ, όργανο ισοδύναμο με τα δύο παραπάνω. Αποθηκεύει σε μορφή std22 ή lst22 και στο πρώτο πεδίο αποθηκεύεται η μέγιστη τιμή και στο δεύτερο η ελάχιστη. Αντί αυτού του οργάνου προτιμάται σαφώς η χρήση των δύο ξεχωριστών οργάνων.
- **ΘΡΜΥΓΡΜΤ, ΘΡΜΥΓΡΓΡ.** Είναι το ΘΕΡΜΟΥΓΡΟΜΕΤΡΟ και ο ΘΕΡΜΟΥΓΡΟΓΡΑΦΟΣ που μετρούν ΘΕΡΜΟΚ-Υ.

3.9.2.5 Θερμοκρασία εδάφους. Η θερμοκρασία εδάφους, ΘΡΜΕΔΑΦ, μετριέται απευθείας στα 6 βάθη: 0, 2,5, 10, 20, 50, 100 cm. Συνεπώς κάθε μέτρηση της περιλαμβάνει ταυτόχρονα 6 τιμές. Η αποθήκευση γίνεται σε μορφή std6 ή lst6 σε μονάδες βαθμών C με ακρίβεια 1 δεκαδικό γρηφίο.

- **ΘΡΜΕΔΑΦ, ΘΡΜΕΔΑΦΜ, ΘΡΜΕΔΑΦΕ.** Είναι τα όργανα ΘΕΡΜΟΜΕΤΡΟ ΕΔΑΦΟΥΣ, ΘΕΡΜΟΜΕΤΡΟ ΕΔΑΦΟΥΣ ΜΕΓΙΣΤΩΝ και ΘΕΡΜΟΜΕΤΡΟ ΕΔΑΦΟΥΣ ΕΛΑΧΙΣΤΩΝ. Τα δύο τελευταία μετρούν τις μέγιστες και ελάχιστες θερμοκρασίες στα 6 βάθη, κυρίως όμως ενδιαφέρει η πρώτη από αυτές, αυτή του "γυμνού εδάφους" σε βάθος 0.

3.9.2.6 Σχετική υγρασία. Τα όργανα που μετρούν τη σχετική υγρασία, ΣΧΕΤ-ΥΓΡ, έχουν ακρίβεια 0 δεκαδικών και μετρούν σε %, αποθηκεύουν δε σε std1 και 1st1 τη μετρηθείσα τιμή στο πρώτο πεδίο.

- ΥΓΡΜΤΡ, ΥΓΡΓΡ, ΥΓΡΜΤΡΑΣ. Είναι τα όργανα ΥΓΡΟΜΕΤΡΟ, ΥΓΡΟΓΡΑΦΟΣ και ΥΓΡΟΜΕΤΡΟ ASSMAN.

3.9.2.7 Δρόσος. Τα όργανα μέτρησης της μεταβλητής ΔΡΟΣΟΣ μετρούν σε gr με ακρίβεια 1 δεκαδικό και αποθηκεύουν σε μορφή std2 ή 1st2 την μετρηθείσα τιμή σε ένα πεδίο.

- ΔΡΕΜΤΡ, ΔΡΕΓΡ. Είναι τα όργανα ΔΡΟΣΟΜΕΤΡΟ και ΔΡΟΣΟΓΡΑΦΟΣ.

3.9.2.8 Άνεμος. Ο ΑΝΕΜΟΣ είναι μια μεταβλητή που μετράται από πολλά διαφορετικά όργανα. Χαρακτηριστικό τους είναι το ό,τι ίσως να πρέπει να συνδυάζονται οι πληροφορίες τους για εξαγωγή ορθών συμπερασμάτων (πχ. ριπή ανέμου).

- ΑΝΜΜΤΡ, ΑΝΜΓΡ. Τα όργανα ΑΝΕΜΟΜΕΤΡΟ και ΑΝΕΜΟΓΡΑΦΟΣ μετρούν διεύθυνση και ταχύτητα ανέμου και τα αποθηκεύουν σε μορφή std11 ή 1st11, τη διεύθυνση στο πρώτο πεδίο και την ταχύτητα στο δεύτερο. Η διεύθυνση αποθηκεύεται σε δεκάδες μοίρες (οπότε μια τιμή πχ. 18 στο πρώτο πεδίο σημαίνει 180 μοίρες) και άρα μπορεί να θεωρηθεί πως η μονάδα μέτρησης είναι οι μοίρες και η ακρίβεια το -1 (οπότε πχ. $18 / 10^{-1} = 180$). Για την ταχύτητα η μονάδα είναι τα m/sec (τυχόν απαίτηση παρουσίασης σε κόμβους ή Beaufort πρέπει να ικανοποιείται στο επίπεδο της εφαρμογής) και η ακρίβεια 0 δεκαδικά ψηφία.
- ΑΝΜΜΤΡΜ. Το ΑΝΕΜΟΜΕΤΡΟ ΜΕΓΙΣΤΩΝ μετρά ριπή ανέμου { Γ } με ίδια κατά τ' άλλα χαρακτηριστικά με το ΑΝΜΜΤΡ. Συνδυάζεται πάντα με μέτρηση ΑΝΜΜΤΡ όπως στο παράδειγμα:

263, 5/12/1994 06:00, ..., 7, 8

Αν αυτή είναι η μέτρηση του ΑΝΜΜΤΡ, δηλαδή 70 μοίρες και 8 m/sec, τότε η μέτρηση του ΑΝΜΜΤΡΜ μπορεί να είναι:

264, 5/12/1994 06:00, ..., 6, 10

δηλαδή στο ίδιο χρονικό διάστημα στο οποίο αναφέρεται η μέτρηση του ΑΝΜΜΤΡ η ριπή ανέμου που μετρήθηκε από το ΑΝΜΜΤΡΜ ήταν 10 m/sec με διεύθυνση 60 μοίρες.

- ΑΝΜΔΕΙΚΤ. Ο ΑΝΕΜΟΔΕΙΚΤΗΣ αποθηκεύει σε std11 ή 1st11 με ακρίβεια -1 και μονάδα τις μοίρες τη διεύθυνση του ανέμου.
- ΑΘΑΝΜΜΤΡ. Το ΑΘΡΟΙΣΤΙΚΟ ΑΝΕΜΟΜΕΤΡΟ μετρά τη συνολική "ποσότητα" ή μήκος διαδρομής ανέμου για τη χρονική περίοδο μέτρησης. Μετρά σε εκατοντάδες μέτρα, οπότε η μονάδα μέτρησης είναι τα m και η ακρίβεια είναι -2. Η μορφή αποθήκευσης είναι η std4 ή η 1st4.

- **ΑΝΜΜΤΡ-Σ.** Το ΣΗΜΑΝΤΙΚΟ ΑΝΕΜΟΜΕΤΡΟ μετρά σημαντικές αποκλίσεις στη διεύθυνση του ανέμου. Αποθηκεύεται σε μορφή std11 ή lst11 σε μοίρες με ακρίβεια -1. Το πρώτο πεδίο είναι η "ελάχιστη" μετρηθείσα διεύθυνση και το δεύτερο η "μέγιστη". Συνδυάζεται πάντα με μέτρηση ανεμόμετρου, όπως στο παράδειγμα:

263, 5/12/1994 06:00, ..., 7, 8

Τότε η μέτρηση του ΑΝΜΜΤΡ-Σ μπορεί να έχει τη μορφή

265, 5/12/1994 06:00, ..., 2, 11

που σημαίνει πως η διεύθυνση του ανέμου κυμάνθηκε από 20 ως 110 μοίρες για το ίδιο χρονικό διάστημα στο οποίο αναφέρεται η μέτρηση του ΑΝΜΜΤΡ. Τυχόν "σημαντική" ένταση του ανέμου για το ίδιο διάστημα φαίνεται από μετρήσεις του ΑΝΜΜΤΡΜ.

- **ΑΝΜΜΤΡ-Δ.** Το ΔΙΑΤΜΗΤΙΚΟ ΑΝΕΜΟΜΕΤΡΟ μετρά το διατμητικό άνεμο σε μια ορισμένη περιοχή. Ο άνεμος αυτός αποθηκεύεται σε μορφή std11 ή lst11 και είναι στο πρώτο πεδίο η ένταση του ανέμου στην απογείωση και στο δεύτερο η ένταση του ανέμου στην προσγείωση. Και στις δύο περιπτώσεις για ύψος 500 m (1600 ft) με ακρίβεια 0 και μονάδες m/sec.

3.9.2.9 Στοιχεία ποιότητας σταθμών ΥΥΥ. Οι σταθμοί ΥΥΥ μετρούν διάφορα στοιχεία ποιότητας νερού, δηλαδή τη συγκέντρωση χημικών στοιχείων στο νερό της πηγής ή της γεώτρησης. Αυτές οι μετρήσεις αποθηκεύονται σε μορφή std2 ή lst2 με ακρίβεια γενικά 2 δεκαδικών ψηφίων και μονάδα μέτρησης το mg/l, εκτός ορισμένων εξαιρέσεων. Είναι συγκεκριμένα:

- M-SIO₂, M-CO₂, M-H₂S, M-KA, M-NA, M-CA, M-MG, M-FE, M-MN, M-NH₄, M-AL, M-CL, M-HCO₃, M-CO₃, M-NO₃, M-NO₂, M-SO₄, M-F, M-PO₄, M-AG, M-AS, M-CD, M-CR, M-HG, M-CU, M-NI, M-PB, M-ZN, M-ΕΗΡΟΥΠ: Πρόκειται για ΜΕΤΡΗΣΗ ΔΙΟΞΕΙΔΙΟΥ ΠΥΡΙΤΙΟΥ, ΜΕΤΡΗΣΗ ΔΙΟΞΕΙΔΙΟΥ ΑΝΘΡΑΚΑ, ΜΕΤΡΗΣΗ ΥΔΡΟΘΕΙΟΥ, ΜΕΤΡΗΣΗ ΚΑΛΙΟΥ, ΜΕΤΡΗΣΗ ΝΑΤΡΙΟΥ, ΜΕΤΡΗΣΗ ΑΣΒΕΣΤΙΟΥ, ΜΕΤΡΗΣΗ ΜΑΓΝΗΣΙΟΥ, ΜΕΤΡΗΣΗ ΣΙΔΗΡΟΥ, ΜΕΤΡΗΣΗ ΜΑΓΓΑΝΙΟΥ, ΜΕΤΡΗΣΗ ΑΜΜΩΝΙΑΣ, ΜΕΤΡΗΣΗ ΑΡΓΙΛΙΟΥ, ΜΕΤΡΗΣΗ ΧΛΩΡΙΟΥ, ΜΕΤΡΗΣΗ ΟΞΙΝΩΝ ΑΝΘΡΑΚΙΚΩΝ, ΜΕΤΡΗΣΗ ΑΝΘΡΑΚΙΚΩΝ, ΜΕΤΡΗΣΗ ΝΙΤΡΙΚΩΝ, ΜΕΤΡΗΣΗ ΝΙΤΡΩΔΩΝ, ΜΕΤΡΗΣΗ ΘΕΙΙΚΩΝ, ΜΕΤΡΗΣΗ ΦΘΟΡΙΟΥ, ΜΕΤΡΗΣΗ ΦΩΣΦΟΡΙΚΩΝ, ΜΕΤΡΗΣΗ ΑΡΓΙΛΟΥ, ΜΕΤΡΗΣΗ ΑΡΣΕΝΙΚΟΥ, ΜΕΤΡΗΣΗ ΚΑΔΜΙΟΥ, ΜΕΤΡΗΣΗ ΧΡΩΜΙΟΥ, ΜΕΤΡΗΣΗ ΥΔΡΑΡΓΥΡΟΥ, ΜΕΤΡΗΣΗ ΧΑΛΚΟΥ, ΜΕΤΡΗΣΗ ΝΙΚΕΛΙΟΥ, ΜΕΤΡΗΣΗ ΜΟΛΥΒΔΟΥ, ΜΕΤΡΗΣΗ ΨΕΥΔΑΡΓΥΡΟΥ και ΜΕΤΡΗΣΗ ΕΗΡΟΥ ΥΠΟΛΛΕΙΜΑΤΟΣ. Έτσι μετρούνται αντίστοιχα οι μεταβλητές Δ-ΠΥΡΙΤ, Δ-ΑΝΘΡ, ΥΔΡΟΘΕΙΟ, ΚΑΛΙΟ, ΝΑΤΡΙΟ,

ΑΣΒΕΣΤΙΟ, ΜΑΓΝΗΣΙΟ, ΣΙΔΗΡΟΣ, ΜΑΓΓΑΝΙΟ, ΑΜΜΩΝΙΑ, ΑΡΓΙΛΙΟ, ΧΛΩΡΙΟ, ΟΞ-ΑΝΘΡ, ΝΙΤΡΙΚΑ, ΝΙΤΡΩΔΗ, ΘΕΙΙΚΑ, ΦΘΟΡΙΟ, ΦΩΣΦΟΡΚΑ, ΑΡΓΙΛΟΣ, ΑΡΣΕΝΙΚΟ, ΚΑΔΜΙΟ, ΧΡΩΜΙΟ, ΥΔΡΑΡΓΥΡ, ΧΑΛΚΟΣ, ΝΙΚΕΛΙΟ, ΜΟΛΥΒΔΟΣ, ΨΕΥΔΑΡΓΡ και ΞΗΡΟΥΠΟΛ.

- **Μ-Τ:** Η ΜΕΤΡΗΣΗ ΘΕΡΜΟΚΡΑΣΙΑΣ του νερού του σταθμού -μεταβλητή ΘΕΡΜΟΚΡΕ- γίνεται σε βαθμούς C.
- **Μ-ΡΗ:** Η ΜΕΤΡΗΣΗ Ρ-Η -μεταβλητή ΡΗ- δεν έχει μονάδες.
- **Μ-ΑΓΩΓΙΜ:** Η ΜΕΤΡΗΣΗ ΑΓΩΓΙΜΟΤΗΤΑΣ -μεταβλητή ΑΓΩΓΙΜΟΤ- γίνεται σε $\mu\text{S}/\text{cm}$.
- **Μ-ΕΚΛΗΡΤ:** Η ΜΕΤΡΗΣΗ ΣΚΛΗΡΟΤΗΤΑΣ -μεταβλητή ΣΚΛΗΡΤΗΤ- γίνεται σε D.
- **Μ-ΔΥΝΟΞΔ:** Η ΜΕΤΡΗΣΗ ΔΥΝΑΜΙΚΟΥ ΟΞΕΙΔΟΑΝΑΓΩΓΗΣ -μεταβλητή ΔΥΝΟΞΕΙΔ- έχει μονάδα το Volt.

3.9.2.10 Δοκιμαστικές αντλήσεις σταθμών ΥΥΥ. Οι δοκιμαστικές αντλήσεις των σταθμών ΥΥΥ, ΔΟΚ-ΑΝΤΛ, αποθηκεύονται σε μορφή std344. Η μορφή αυτή αποτελείται από 8 πεδία.

- **ΔΟΚ-ΑΝΤΛ:** Η ΔΟΚΙΜΑΣΤΙΚΗ ΑΝΤΛΗΣΗ αποθηκεύεται στον πίνακα ως εξής: στο integer1 πρώτο πεδίο, value0, αποθηκεύεται ο κώδικας της αντλίας που χρησιμοποιήθηκε. Ο πίνακας αναφοράς για τους κώδικες, codes, προσπελάσσεται με είδος κώδικα, code_ts, ΑΝΤΛΙΑ. Δεν χρησιμοποιείται κάποιος ιδιαίτερος κώδικας και άρα εισάγεται κωδικοποίηση του ΥΔΡΟΣΚΟΠΙΟΥ. Στο πεδίο value1, τύπου integer4, αποθηκεύεται η παροχή της άντλησης σε m^3/h και στο integer4 πεδίο value2 η στάθμη της γεώτρησης τη στιγμή της άντλησης σε m. Και στα δύο αυτά πεδία η ακρίβεια είναι 2 δεκαδικά. Στα integer2 πεδία value3 και value4 αποθηκεύονται οι χρόνοι άντλησης και επαναφοράς σε min. Τα υπόλοιπα πεδία, value5 - 7, τύπου integer1 είναι δεσμευμένα για μελλοντική χρήση.

3.9.2.11 Εξάτμιση. Η εξάτμιση, ΕΞΑΤΜΙΣΗ, μετρείται σε mm με ακρίβεια 1 δεκαδικό και αποθηκεύεται με μορφή std2 ή 1st2. Στις περιπτώσεις που η εξάτμιση δεν μετράται κατευθείαν αλλά υπολογίζεται από βοηθητικές παραμέτρους, αυτός ο υπολογισμός πρέπει να γίνεται από το πρόγραμμα εισαγωγής. Δηλαδή η εξάτμιση δεν θεωρείται παράγωγο όργανο. Αντίθετα, παράγωγο όργανο είναι βέβαια η εξατμισοδιαπνοή.

- **ΕΞΤΜΜΤΡ, ΕΞΤΜΜΓΡ, ΕΞΤΜΜΤΡΡ, ΕΞΤΜΜΤΡΛ:** Πρόκειται για τα όργανα ΕΞΑΤΜΙΣΗΜΕΤΡΟ, ΕΞΑΤΜΙΣΗΓΡΑΦΟΣ, ΕΞΑΤΜΙΣΗΜΕΤΡΟ ΡΙΧΗ και ΕΞΑΤΜΙΣΗΜΕΤΡΟ ΛΕΚΑΝΗΣ.
- **ΕΞΑΤΜΔΠΝ:** Η ΕΞΑΤΜΙΣΟΔΙΑΠΝΟΗ είναι παράγωγο όργανο.

3.9.2.12 Ατμοσφαιρική πίεση. Η ατμοσφαιρική πίεση, ΠΙΕΣΗ, μετριέται καταρχήν απευθείας από κάποιο είδος βαρόμετρου και στη συνέχεια ανάγεται σε πίεση στο επίπεδο της θάλασσας. Μετριέται πάντα σε mbar με ακρίβεια 1 δεκαδικό ψηφίο.

- **ΒΑΡΜΤΡ, ΒΑΡΓΡ.** Πρόκειται για ΒΑΡΟΜΕΤΡΟ, ΒΑΡΟΓΡΑΦΟ που αποθηκεύουν σε μορφή std22 ή lst22. Το πρώτο πεδίο, value0 τύπου integer2 είναι η μετρηθείσα πίεση και το δεύτερο η αντίστοιχη θερμοκρασία -για το πεδίο αυτό ισχύουν όσα και για τις συνήθεις μετρήσεις θερμοκρασίας.
- **ΠΙΕΣΗ.** Η ΠΙΕΣΗ είναι ένα παράγωγο όργανο. Αποθηκεύει σε μορφή std211 με τον ακόλουθο τρόπο: στο πρώτο πεδίο, value0, τύπου integer2 αποθηκεύεται η ανηγμένη σε επίπεδο θάλασσας πίεση. Η τιμή αυτή παράγεται από την αντίστοιχη τιμή του ΒΑΡΜΤΡ. Στο δεύτερο πεδίο, value1, τύπου integer1 αποθηκεύεται η βαρομετρική τάση, δηλαδή η αριθμητική μεταβολή της πίεσης κατά το προηγούμενο χρονικό διάστημα. Η τάση αποθηκεύεται σε mbar με ακρίβεια 1, οπότε μπορεί να πάρει τιμές από -12.7 ως 12.7 mbar. Τέλος, το σύμβολο της βαρομετρικής τάσης αποθηκεύεται στο integer1 τρίτο πεδίο value2 ως κώδικας (code_t = 'ΠΙΕΣΗ') WMO 0200.

3.9.2.13 Ηλιοφάνεια και ακτινοβολία. Η ηλιοφάνεια, ΗΛΙΟΦΑΝ, και η ακτινοβολία, ΑΚΤΙΝΟΒ, αποθηκεύονται σε ένα πεδίο που περιέχει την τιμή της μέτρησης.

- **ΑΚΤΙΝΓΡ, ΑΚΤΙΝΜΤΡ, ΦΙΛΤΡΑΚΤ:** Ο ΑΚΤΙΝΟΓΡΑΦΟΣ, το ΑΚΤΙΝΟΜΕΤΡΟ και τα ΦΙΛΤΡΑ ΑΚΤΙΝΟΒΟΛΙΑΣ αποθηκεύουν σε μορφή std2 ή lst2 και μετρούν την ακτινοβολία σε W/m^2 με ακρίβεια 0 δεκαδικά ψηφία.
- **ΑΚΤΙΝΓΡΜ:** Ο ΑΚΤΙΝΟΓΡΑΦΟΣ ΜΕΓΑΛΟΥ ΜΗΚΟΥΣ ΚΥΜΑΤΟΣ είναι παράγωγο όργανο. Μετρά σε W/m^2 με 0 δεκαδικά ψηφία ακρίβεια σε μορφή std1 ή lst1.
- **ΗΛΙΟΓΡ:** Ο ΗΛΙΟΓΡΑΦΟΣ μετρά τις ώρες ηλιοφάνειας κατά το χρονικό διάστημα της μέτρησης με ακρίβεια 1 δεκαδικό ψηφίο. Αποθηκεύεται σε μορφή std2 ή lst2.

3.9.2.14 Νέφωση. Η νέφωση, ΝΕΦΩΣΗ, είναι όργανο του οποίου οι μετρήσεις προέρχονται από ανθρώπινη παρατήρηση και αφορούν στο είδος και το ύψος των χαμηλών, μέσων και υψηλών νεφών και το ολικό ποσοστό νεφοκάλυψης.

- **ΝΕΦΩΣΗ:** Η ΝΕΦΩΣΗ αποθηκεύεται σε μορφή std9. Όλα τα πεδία εκτός του value1 είναι τύπου integer1. Όλα τα ποσά νεφοκάλυψης εκφράζονται σε Ογδοα με ακρίβεια 0 (παίρνουν τιμές από 0, δηλαδή 0/8, ως 8, δηλαδή 8 / 8). Όλα τα είδη νεφών (γένος και τύπος τους) αποθηκεύονται ως κώδικας με είδος κώδικα code_t ΝΕΦΩΣΗ. Ο χρησιμοποιούμενος κώδικας είναι ο WMO 0500 για το γένος του νέφους. Ωστόσο υπάρχουν και περισσότεροι λεπτομερείς κώδικες που περιγράφουν επακριβώς το νέφος. Για το λόγο αυτό εισάγεται η εξής μεθοδολογία: Ο κώδικας

WMO 0500 έχει γένη νεφών με αντίστοιχες τιμές κώδικα από 0 ως 9. Η τιμή "7" του κώδικα αντιστοιχίζεται στην τιμή -1 στη ΒΔ (δεν πρέπει να χρησιμοποιηθεί τιμή null διότι σημαίνει έλλειψη μέτρησης ενώ η τιμή "7" του κώδικα σημαίνει παρουσία μέτρησης αλλά αδυναμία παρατήρησης νεφών). Σε περίπτωση λοιπόν που η αποθηκευμένη τιμή στη ΒΔ είναι μικρότερη από 10, αυτό σημαίνει πως χρησιμοποιείται ο κώδικας WMO 0500, απλά δηλαδή επισημαίνεται το γένος των νεφών. Αν η τιμή στη ΒΔ είναι μεγαλύτερη ή ίση του 10, αυτό σημαίνει πως υπάρχει ειδικότερη περιγραφή του γένους και του τύπου του νέφους σύμφωνα με τους κώδικες WMO 0509 (για νέφη CIRRUS, CIRROCUMULUS και CIRROSTRATUS με κωδικούς WMO 0500 αντίστοιχα 0, 1 και 2), WMO 0513 (για νέφη STRATOCUMULUS, STRATUS, CUMULUS και CUMULONIMBUS με κωδικούς WMO 0500 αντίστοιχα 6, 7, 8 και 9) και WMO 0515 (για νέφη ALTOCUMULUS, ALTOSTRATUS και NIMBOSTRATUS με κωδικούς WMO 0500 3, 4 και 5 αντίστοιχα). Η τιμή που αποθηκεύεται τότε είναι η τιμή του κώδικα WMO 0500 πολλαπλασιασμένη επί 10 συν την τιμή του αντίστοιχου κώδικα για το συγκεκριμένο γένος νεφών. Για παράδειγμα: αν αποθηκεύεται η τιμή 6 (κώδικας WMO 0500) αυτό σημαίνει πως το νέφος είναι γένους STRATOCUMULUS. Αν όμως αποθηκεύεται η τιμή 64 ($= 6 * 10 + 4$) αυτό σημαίνει πως το γένος είναι STRATOCUMULUS και ειδικότερα από τον κώδικα WMO 0513 που αναφέρεται σε αυτό το γένος νεφών, STRATOCUMULUS CUMULOGENITUS (τιμή 4 του κώδικα WMO 0513). Αντίστοιχα, η τιμή 3 σημαίνει νέφη ALTOCUMULUS (κώδικας WMO 0500) ενώ η 39 ($= 3 * 10 + 9$) σημαίνει ALTOCUMULUS χαώδους ουρανού (τιμή 9 κώδικα WMO 0515). Ειδικά για τα νέφη CIRRUS με κωδικό 0 αυτή η μεθοδολογία θα οδηγούσε σε σύγκρουση και αβεβαιότητα τιμών μια και $0 * 10 = 0$ οπότε δεν θα ήταν γνωστό αν τα νέφη είναι σύμφωνα με τον κωδικό WMO 0500 ή είναι ειδικότερη περιγραφή CIRRUS. Για το λόγο αυτό τα CIRRUS αποθηκεύονται είτε σύμφωνα με τον κώδικα WMO 0500 ως 0 είτε με τιμές $100 +$ την τιμή του κώδικα WMO 0509. Για παράδειγμα 0 σημαίνει CIRRUS και 103 σημαίνει CIRRUS SPISSATUS CUMULONIMBOGENITUS (τιμή 3 του κώδικα WMO 0509). Σε κάθε περίπτωση ο χρήστης δεν χρειάζεται να απασχοληθεί με αυτές τις λεπτομέρειες, μια και ο πίνακας codes περιέχει όλες τις ορθές τιμές και οι εφαρμογές μπορούν να παρουσιάζουν αυτόματα (όπως περιγράφηκε στην παράγραφο 3.9.2.2) την ορθή περιγραφή (codes.descr) και όχι την αριθμητική τιμή (codes.code για codes.code_t = 'ΝΕΦΩΣΗ') του κώδικα. Προσοχή χρειάζεται μόνο στα προγράμματα εισαγωγής τιμών νέφωσης στη ΒΔ ώστε να λαμβάνονται υπόψη αυτές οι κωδικοποιήσεις.

Το πεδίο value0 περιέχει την ολική νεφοκάλυψη. Το πεδίο value1 περιέχει το ύψος σε m με ακρίβεια 0 δεκαδικά της βάσης του πρώτου χαμηλού στρώματος νεφών, το πεδίο value2 το ποσό της νεφοκάλυψης του πρώτου χαμηλού στρώματος νεφών

και το πεδίο value3 το είδος (γένος - τύπος) των νεφών του πρώτου χαμηλού στρώματος. Το πεδίο value4 αποθηκεύει το ποσό της νεφοκάλυψης του δεύτερου χαμηλού στρώματος νεφών και το πεδίο value5 το είδος τους. Το πεδίο value6 είναι το ποσό της νεφοκάλυψης του μεσαίου στρώματος και το value7 το είδος των νεφών. Τέλος το value8 περιλαμβάνει το είδος των νεφών του υψηλού στρώματος.

3.9.2.15 Κατάσταση εδάφους. Η κατάσταση του εδάφους -μεταβλητή ΚΑΤ-ΕΔΑΦ- προέρχεται από ανθρώπινη παρατήρηση και αποθηκεύεται ως κώδικας στο πρώτο πεδίο σε μορφή std1 ή lst1.

- **ΕΔΑΦΟΣ:** Το όργανο ΕΔΑΦΟΣ αποθηκεύει με ακρίβεια 0 δεκαδικών έναν κώδικα (code_t = 'ΕΔΑΦΟΣ'). Ο κώδικας που χρησιμοποιείται είναι ο WMO 0901. Σε περίπτωση που οι συνθήκες υπόβαθρου υποδεικνύουν, μέσω των πεδίων S ή I της λέξης κατάστασης, πως υπήρχε χιόνι ή πάγος κατά τη στιγμή της μέτρησης, τότε ο κώδικας που χρησιμοποιείται είναι ο WMO 0975 αυξημένος κατά 10, ώστε να μην υπάρχει σύγκρουση ή αβεβαιότητα τιμών. Δηλαδή τιμή 1 σημαίνει, σύμφωνα με τον WMO 0901, ΕΠΙΦΑΝΕΙΑ ΕΔΑΦΟΥΣ ΥΓΡΗ ενώ τιμή 10 (= 0 + 10) σημαίνει ΕΔΑΦΟΣ ΚΑΛΥΜΕΝΟ ΑΠΟ ΠΑΓΟ (όπου 0 η τιμή του κώδικα WMO 0975 και αντίστοιχα το πεδίο I της λέξης κατάστασης να είναι 1). Για αυτό χρειάζεται ιδιαίτερη μέριμνα από τα προγράμματα εισαγωγής ενώ η παρουσίαση γίνεται, και πάλι, αυτόματα.

3.9.2.16 Ορατότητα. Η μέτρηση ορατότητας -μεταβλητή ΟΡΑΤΟΤΗΣ- προέρχεται από ανθρώπινη παρατήρηση και έχει δυο διαφορετικά όργανα.

- **ΟΡΑΤΟΤΗΣ.** Το όργανο ΟΡΑΤΟΤΗΣ μετρά σε m με ακρίβεια 0 δεκαδικών σε μορφή std4 ή lst4. Το integer4 πεδίο value0 περιέχει την τιμή της ορατότητας. Τιμές ορατότητας πάνω από 100000 ή μικρότερες από 0 σημαίνουν "ορατότητα απεριόριστη" (CAVOK). Η μέτρηση αναφέρεται ταυτόχρονα σε όλα τα σημεία του ορίζοντα.
- **ΟΡΑΤΟΤ-Μ.** Το όργανο ΟΡΑΤΟΤΗΣ METAR μετρά την ορατότητα στις 8 κύριες διευθύνσεις (ανά 45 μοίρες), είναι δηλαδή πιο "λεπτομερές" από το ΟΡΑΤΟΤΗΣ. Αποθηκεύει τις μετρήσεις σε m με ακρίβεια 0 δεκαδικών και με τιμές πάνω από 100000 ή μικρότερες από 0 να σημαίνουν CAVOK σε μορφή lst4 όπου όλα τα πεδία είναι τύπου integer4. Συγκεκριμένα το πεδίο value0 περιέχει τη γενική ορατότητα (αντίστοιχο δηλαδή με το όργανο ΟΡΑΤΟΤΗΣ), το value1 την ορατότητα στις 0 μοίρες, το value2 στις 45, το value3 στις 90 και αντίστοιχα μέχρι το value8 που περιέχει την ορατότητα στις 315 μοίρες. Το πεδίο value9 περιέχει την κατακόρυφη ορατότητα.

3.9.2.17 Τάση υδρατμών. Η τάση υδρατμών, ΤΑΣΥΔΡ, μετριέται σε mbar με ακρίβεια 0 και αποθηκεύεται σε μορφή std1 ή lst1.

- ΤΑΣΥΔΡ, ΤΑΣΚΡΥΔΡ: Πρόκειται για τα παράγωγα όργανα ΤΑΣΗ ΥΔΡΑΤΜΩΝ και ΤΑΣΗ ΚΟΡΕΣΜΕΝΩΝ ΥΔΡΑΤΜΩΝ.

3.9.2.18 Ραδιοβολίσεις. Κάθε ραδιοβολίση είναι μια σύνθετη διαδικασία μέτρησης μετεωρολογικών μεταβλητών στα ανώτερα στρώματα της ατμόσφαιρας. Κατάλληλα μπαλόνια με μετρητικά όργανα στέλνονται στην ατμόσφαιρα και διαβιβάζουν με ασύρματο τις μετρήσεις τους. Οι μετρήσεις γίνονται για τις μεταβλητές ΘΕΡΜΟΚ, ΑΝΕΜΟΣ, ΣΧΕΤ-ΥΓΡ, ΥΨΟΜΕΤΡΟ και διαβιβάζονται καταρχήν σε προκαθορισμένες στάθμες πίεσης. Δηλαδή, όταν το μπαλόνι φτάσει σε κάποιο ύψος στο οποίο η ατμοσφαιρική πίεση έχει μια από τις συγκεκριμένες προκαθορισμένες τιμές, διαβιβάζονται στη βάση οι τιμές της θερμοκρασίας, του ανέμου, της σχετικής υγρασίας και το υγόμετρο σε αυτή τη στάθμη πίεσης. Στην επόμενη προκαθορισμένη στάθμη διαβιβάζονται πάλι οι μετρήσεις κοκ. Τυχόν εξαιρετικές τιμές κάποιας μεταβλητής ανάμεσα σε δύο στάθμες πίεσης επίσης διαβιβάζονται. Τέλος γίνονται και μετρήσεις για την τροπόπαυση. Οι προκαθορισμένες στάθμες πίεσης είναι 32, οι ακόλουθες (τιμές σε mbar): Επιφάνεια - 1000 - 950 - 900 - 850 - 800 - 750 - 700 - 650 - 600 - 550 - 500 - 450 - 400 - 350 - 300 - 250 - 200 - 175 - 150 - 125 - 100 - 80 - 70 - 60 - 50 - 40 - 30 - 20 - 15 - 10 - 7.

Στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ οι ραδιοβολίσεις αποθηκεύονται με σύνθετο τρόπο. Κάθε σταθμός που εκτελεί ραδιοβολίσεις περιλαμβάνει και μια σειρά οργάνων, κάθε ένα από τα οποία μετρά κάποια από τις μεταβλητές της ραδιοβολίσης κατά τη διάρκειά της. Όλα τα όργανα αποθηκεύουν σε μορφή lst όπου στις 32 διαδέσιμες τιμές της μορφής αποθηκεύονται οι 32 μετρήσεις του οργάνου στις 32 προκαθορισμένες στάθμες. Η ημερομηνία που άρχισε η ραδιοβολίση είναι αυτή που αποθηκεύεται στο πεδίο date της μορφής σε όλες τις εγγραφές. Επίσης οι εξαιρετικές περιπτώσεις τιμών καταγράφονται από τα "σημαντικά" όργανα. Συνεπώς μια ραδιοβολίση σε κάποιο σταθμό έχει ως αποτέλεσμα 4 τουλάχιστον "ταυτόχρονες" εγγραφές 32 τιμών στα αντίστοιχα όργανα (ΡΔΘΡΜ, ΡΔΑΝΜ ή ΡΔΑΝΜ-Ρ, ΡΔΥΓΡ, ΡΔΥΨΟΜ), μια τουλάχιστον για την πίεση αναφοράς στην επιφάνεια (ΡΔΠΙΕΣΗ) και πιθανά και στα υπόλοιπα (ΡΔΤΡΟΠ και τα "σημαντικά" όργανα). Τα όργανα που χρησιμοποιούνται είναι:

- ΡΔΘΡΜ Το ΡΑΔΙΟΘΕΡΜΟΜΕΤΡΟ μετρά ΘΕΡΜΟΚ στις 32 διαφορετικές στάθμες σε βαθμούς C με ακρίβεια 1 δεκαδικό ψηφίο. Αποθηκεύει σε μορφή lst2 με integer2 πεδία. Δηλαδή την τιμή της επιφάνειας στο value0, την τιμή των 1000 mbar στο value1, την τιμή των 950 mbar στο value2 και την τιμή των 7 mbar στο value31. Αντίστοιχα γίνεται η αποθήκευση για όλα τα όργανα ραδιοβολίσης.
- ΡΔΑΝΜ Το ΡΑΔΙΟΑΝΕΜΟΜΕΤΡΟ μετρά τη μεταβλητή ΑΝΕΜΟΣ στις 32 στάθμες πίεσης. Αποθηκεύει σε μορφή lst11. Όλα τα πεδία είναι τύπου integer1. Στο

πρώτο πεδίο κάθε μέτρησης (value0, value2, value4, value6, ..., value62) αποθηκεύεται η διεύθυνση του ανέμου σε Μοίρες με ακρίβεια -1 (δηλαδή δεκάδες Μοίρες) για τις στάθμες πίεσης (Επιφάνεια, 1000, 950, 900, ..., 7). Στο δεύτερο πεδίο κάθε μέτρησης (value1, value3, value5, value7, ..., value63) αποθηκεύεται η ταχύτητα του ανέμου σε m/sec με ακρίβεια 0 δεκαδικά για τις ίδιες στάθμες πίεσης.

- **ΡΔΥΓΡ.** Το ΡΑΔΙΟΥΓΡΟΜΕΤΡΟ μετρά τη ΣΧΕΤ-ΥΓΡ στις 32 στάθμες πίεσης με ακρίβεια 0 δεκαδικών σε %. Αποθηκεύει σε μορφή 1st1.
- **ΡΔΥΨΟΜ.** Το ΡΑΔΙΟΥΨΟΜΕΤΡΟ μετρά ΥΨΟΜΕΤΡΟ, δηλαδή το υγόμετρο στο οποίο παρουσιάζονται οι 32 στάθμες πίεσης. Αποθηκεύει σε μορφή 1st4 σε m με ακρίβεια 0 δεκαδικών.
- **ΡΔΑΝΜ-Ρ.** Το ΡΑΔΙΟΑΝΕΜΟΜΕΤΡΟ PILOT μετρά επίσης τη μεταβλητή ΑΝΕΜΟΣ. Κατά τ' άλλα μετρά και αποθηκεύει με τον ίδιο τρόπο με το ΡΔΑΝΜ. Σε μια ραδιοβόλιση μπορεί να υπάρχουν και τα δύο όργανα ή μόνο το ένα από τα δύο.
- **ΡΔΠΙΕΣΗ.** Το όργανο ΡΑΔΙΟΠΙΕΣΗ μετρά την ΠΙΕΣΗ στην Επιφάνεια. Η πίεση αυτή δεν είναι απαραίτητα γνωστή μέσω των σταθμών και χρειάζεται η καταγραφή της για αναφορά. Η τιμή της πίεσης δίνεται σε mbar και αποθηκεύεται σε μορφή std2 με ακρίβεια 1 δεκαδικό ψηφίο.
- **ΡΔΤΡΟΠΟΠ.** Το όργανο ΡΑΔΙΟΤΡΟΠΟΠΑΥΣΗ μετρά ΤΡΟΠΟΠ, δηλαδή επισημαίνει τα χαρακτηριστικά της τροπόπαυσης που παρατηρούνται στη διάρκεια της ραδιοβόλισης. Είναι σύνθετο όργανο που αποθηκεύει σε μορφή 1st4 (όλα τα πεδία integer4) ως εξής: στο πεδίο value0 αποθηκεύεται ο αριθμός της τροπόπαυσης, μια και είναι δυνατό η ραδιοβόλιση να συναντήσει περισσότερες από μία. Παίρνει δηλαδή τιμές 1, 2, 3, ... Στο value1 αποθηκεύεται το υγόμετρο σε m με ακρίβεια 0 δεκαδικών της παρούσας (από value0) τροπόπαυσης, στο value2 με ακρίβεια 1 δεκαδικού και σε μονάδες mbar η πίεση που μετρήθηκε, στο value3 με ακρίβεια 1 δεκαδικού και σε βαθμούς C η θερμοκρασία, στο value5 η διεύθυνση του ανέμου σε Μοίρες με ακρίβεια -1 και στο value6 η ταχύτητα του ανέμου σε m/sec με ακρίβεια 0. Στο value4 αποθηκεύεται η ατμοσφαιρική πίεση στην οποία μετρήθηκε αυτός ο άνεμος, πάλι σε mbar με ακρίβεια 1. Σημαντικό είναι το γεγονός πως για τον ίδιο συνδυασμό οργάνου - ημερομηνίας μπορεί να υπάρχουν περισσότερες από μια εγγραφές, μια και μπορεί να καταγραφούν περισσότερες από μια τροπόπαυσεις. Αυτό όμως που καταρχήν τις διαφοροποιεί είναι η τιμή του πεδίου value0, δηλαδή ο αριθμός της τροπόπαυσης που όμως δεν είναι φυσικά μέρος του κλειδιού { Γ }. Αν λοιπόν η λέξη κατάστασης είναι η ίδια τότε οι δύο εγγραφές θα θεωρούνται μια από τη ΒΔ. Για το λόγο αυτό πρέπει να χρησιμοποιηθεί σε αυτή την περίπτωση το πεδίο χρήστη (U) της λέξης κατάστασης. Συγκεκριμένα, για την πρώτη τροπόπαυση το πεδίο

θα παίρνει τιμή 0, για τη δεύτερη 1 κοκ. Η ερμηνεία αυτού του πεδίου εξαρτάται, όπως φαίνεται σε αυτή την περίπτωση, από την εφαρμογή και το είδος της αποθηκευόμενης πληροφορίας.

- **ΡΔΘΡΜ-Σ.** Το ΣΗΜΑΝΤΙΚΟ ΡΑΔΙΟΘΕΡΜΟΜΕΤΡΟ μετρά ΘΕΡΜΟΚ αλλά μόνο τυχόν αξιοσημείωτες τιμές της που παρατηρήθηκαν κατά τη ραδιοβόλιση ανάμεσα σε δύο προκαθορισμένες στάθμες πίεσης. Αποθηκεύεται σε μορφή std22, όπου στο integer2 πεδίο value1 με ακρίβεια 1 δεκαδικό σε βαθμούς C περιέχεται η θερμοκρασία που μετρήθηκε και στο επίσης integer2 πεδίο value0 περιέχεται η αντίστοιχη πίεση σε mbar με ακρίβεια 1.
- **ΡΔΑΝΜ-Σ, ΡΔΑΝΜ-ΡΣ.** Το ΣΗΜΑΝΤΙΚΟ ΡΑΔΙΟΑΝΕΜΟΜΕΤΡΟ και το ΣΗΜΑΝΤΙΚΟ ΡΑΔΙΟΑΝΕΜΟΜΕΤΡΟ PILOT μετρούν αξιοσημείωτες τιμές της μεταβλητής ΑΝΕΜΟΣ ανάμεσα σε δύο προκαθορισμένες στάθμες πίεσης. Αποθηκεύονται σε μορφή std211 όπου στο integer2 πεδίο value0 αποθηκεύεται η τιμή της πίεσης σε mbar με ακρίβεια 1 δεκαδικό όπου παρατηρήθηκε η αξιοσημείωτη τιμή και στα integer1 πεδία value1 και value2 καταγράφεται αντίστοιχα η διεύθυνση σε Μοίρες με ακρίβεια -1 (δεκάδες Μοίρες) και η ένταση σε m/sec με ακρίβεια 0.
- **ΡΔΑΝΜ-Δ.** Το ΔΙΑΤΜΗΤΙΚΟ ΡΑΔΙΟΑΝΕΜΟΜΕΤΡΟ μετρά τη μεταβλητή ΑΝΕΜΟΣ και συγκεκριμένα το διατμητικό άνεμο πάνω και κάτω από την τρέχουσα στάθμη πίεσης. Αποθηκεύει σε μορφή std211, όπου στο integer2 πεδίο value0 περιέχεται η ατμοσφαιρική πίεση σε mbar με ακρίβεια 1 δεκαδικού στο σημείο όπου παρατηρείται διατμητικός άνεμος και στα integer1 πεδία value1 και value2 αποθηκεύεται η τιμή της έντασης του διατμητικού ανέμου σε m/sec 500 m κάτω και πάνω αντίστοιχα από τη στάθμη πίεσης που επίσης καταγράφηκε.
- **ΡΔΥΓΡ-Σ.** Το ΣΗΜΑΝΤΙΚΟ ΡΑΔΙΟΥΓΡΟΜΕΤΡΟ μετρά αξιοσημείωτες τιμές της μεταβλητής ΣΧΕΤ-ΥΓΡ. Αποθηκεύεται σε μορφή std21 όπου στο integer2 πεδίο value0 υπάρχει η τιμή της πίεσης σε mbar με ακρίβεια 1 και στο integer1 πεδίο value1 υπάρχει η τιμή της σχετικής υγρασίας σε % με ακρίβεια 0.

Ως παράδειγμα αποθήκευσης ραδιοβόλισης στη ΒΔ δίνεται το ακόλουθο: έστω πως κάποιος σταθμός (με κωδικό ΥΔΡΟΣΚΟΠΙΟΥ 512) εκτελεί ραδιοβολίσεις και άρα έχει τα αντίστοιχα όργανα ραδιοβόλισης ΡΔΘΡΜ (με κωδικό ΥΔΡΟΣΚΟΠΙΟΥ 520), ΡΔΑΝΜ (521), ΡΔΥΓΡ (522), ΡΔΥΨΟΜ (523), ΡΔΤΡΟΠΟΠ (524), ΡΔΘΡΜ-Σ (525), ΡΔΑΝΜ-Σ (526), ΡΔΑΝΜ-Δ (527), ΡΔΥΓΡ-Σ (528) και ΡΔΠΙΕΣΗ (529). Τότε για μια ραδιοβόλιση που άρχισε στις 5/12/1994 06:00 μπορούν να εισαχθούν στους διάφορους πίνακες οι εγγραφές:

- Στον πίνακα raw_1st2 θα εισαχθεί εγγραφή για το ΡΔΘΡΜ. Η λέξη κατάστασης παίρνει τιμή 1 (όπως και στις υπόλοιπες εγγραφές) που σημαίνει απλά πως η εγγραφή

ανήκει στο πρώτο επίπεδο. Έτσι και αλλιώς περαιτέρω επεξεργασία δεν προβλέπεται για τα δεδομένα ραδιοβολίσεων. Μετά τη λέξη κατάστασης ακολουθούν 32 τιμές θερμοκρασίας value0 - value31 (η πραγματική τιμή βρίσκεται βέβαια διαιρώντας με 10^1) στις 32 προκαθορισμένες στάθμες πίεσης:

520, 5/12/1994 06:00, 1, 154, 150, 148, 140, ..., -451, -482

- Στον πίνακα raw_1st11 θα εισαχθεί αντίστοιχη εγγραφή για το ΡΔΑΝΜ. Η εγγραφή αυτή θα έχει 64 πεδία value0 - value63 για τις 32 τιμές πίεσης όπου τα "άρτια" πεδία θα είναι διεύθυνση (η πραγματική τιμή βρίσκεται διαιρώντας με 10^{-1}) και τα "περιττά" ταχύτητα:

521, 5/12/1994 06:00, 1, 4, 3, 5, 2, 5, 7, ..., 12, 4, 10, 2

- Στον πίνακα raw_1st1 θα εισαχθεί αντίστοιχη εγγραφή για το ΡΔΥΓΡ. Η εγγραφή αυτή θα έχει 32 πεδία value0 - value31 για τις 32 τιμές σχετικής υγρασίας.

522, 5/12/1994 06:00, 1, 90, 88, 87, 84, ..., 50, 47

- Στον πίνακα raw_1st4 θα εισαχθεί αντίστοιχη εγγραφή για το ΡΔΥΨΟΜ. Η εγγραφή αυτή θα έχει 32 πεδία value0 - value31 για τις 32 τιμές υγομέτρου στις στάθμες πίεσης:

523, 5/12/1994 06:00, 1, 0, 20, 40, 50, ..., 14340, 14517

- Στον πίνακα raw_1st4 αποθηκεύονται 2 εγγραφές για τις δύο τροποποιήσεις - όργανο ΡΔΤΡΟΠΟΠ- που συνάντησε η βολίδα ανεβαίνοντας. Προσοχή χρειάζεται η τιμή της λέξης κατάστασης. Για την πρώτη τροποποίηση είναι 1 δηλαδή μόνο το πεδίο L του επιπέδου 1 έχει τιμή 1 ενώ για τη δεύτερη έχει τιμή 2049 (δυναμικό 10000000001) που σημαίνει πως το πεδίο L έχει τιμή 1 όπως επίσης και το πεδίο U. Φυσικά οι απόλυτες τιμές της λέξης κατάστασης δεν έχουν σημασία, αφού η επεξεργασία των πεδίων γίνεται από τις αντίστοιχες συναρτήσεις getstatus() και setstatus(). Οι τιμές χρειάζονται βέβαια κατάλληλη διαίρεση για να μετατραπούν στα αντίστοιχα φυσικά μεγέθη:

524, 5/12/1994 06:00, 1, 1, 13700, 21, -500, 20, 9, 4

524, 5/12/1994 06:00, 2049, 2, 15000, 18, -504, 50, 15, 8, 10

- Στον πίνακα raw_std2 εισάγεται μια εγγραφή για την πίεση στην επιφάνεια, ΡΔΠΙΕΣΗ με τιμή 1003.2 mbar:

529, 5/12/1994 06:00, 1, 10032

- Στον πίνακα raw_std211 εισάγεται μια εγγραφή ΡΔΑΝΜ-Δ για διατμητικό άνεμο (5 m/sec - 30 m/sec) που συνάντησε το μπαλόκι ανεβαίνοντας σε στάθμη πίεσης 580 mbar:

527, 5/12/1994 06:00, 1, 5800, 5, 30

- Τέλος στον πίνακα raw_std22 εισάγεται μια εγγραφή για αξιοσημείωτη θερμοκρασία (-100.4) του οργάνου ΡΔΘΡΜ-Σ που συνάντησε το μπαλόνι ανεβαίνοντας σε στάθμη πίεσης 174 mbar:

525, 5/12/1994 06:00, 1, 1740, -1004

3.9.2.19 Υδρομετρήσεις και στεροϋδρομετρήσεις. Οι υδρομετρήσεις αποθηκεύονται επίσης με σύνδετο τρόπο.

- **ΥΔΡΜΤΡ.** Η ΥΔΡΟΜΕΤΡΗΣΗ είναι η άμεση μέτρηση της μεταβλητής ΠΑΡΟΧΗ. Ωστόσο είναι σύνδετη η αποθήκευση της διότι δεν αρκεί να αποθηκευθεί μόνο μια τιμή. Μια υδρομέτρηση αρχίζει καταγράφοντας τη διατομή και πιθανόν τα υδραυλικά χαρακτηριστικά του ποταμού στο σημείο της υδρομέτρησης. Στη συνέχεια λαμβάνονται ενδείξεις σταθμημέτρων και οι ταχύτητες του νερού στα "στοιχειώδη εμβαδά" της διατομής, οπότε ολοκληρώνοντας προκύπτει η παροχή. Για το λόγο αυτό δεν αρκεί η αποθήκευση μόνο της παροχής (οι ταχύτητες δεν αποθηκεύονται στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ) αλλά απαιτείται η αποθήκευση και των μετρήσεων των σταθμημέτρων και των στοιχείων διατομής και υδραυλικών χαρακτηριστικών. Για το σκοπό αυτό κάθε υδρομέτρηση αποθηκεύεται σε 4 διαφορετικούς πίνακες. Στον raw_std34 αποθηκεύεται η καθ' αυτή τιμή της παροχής που μετρήθηκε (προέκυψε από την ολοκλήρωση) με ακρίβεια 3 δεκαδικά και μονάδες m³/sec στο integer4 πεδίο value0. Στο integer4 πεδίο value1 αποθηκεύεται ο α/α της σταθεράς (το πεδίο constant) τύπου ΔΙΑΤΟΜΗ που αναφέρεται στην παρούσα υδρομέτρηση ενώ στο integer4 πεδίο value2 αποθηκεύεται ο α/α της σταθεράς τύπου ΥΔΡ-ΧΑΡ που επίσης αναφέρεται στην παρούσα υδρομέτρηση. Ο εντοπισμός των σταθερών διατομής και υδραυλικών χαρακτηριστικών που αντιστοιχούν στην υδρομέτρηση είναι έτσι απλός: το πεδίο obj0 στον πίνακα αποθήκευσης της σταθεράς θα είναι ίσο με τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ της υδρομέτρησης και το πεδίο constant ίσο με το αποθηκευμένο στο αντίστοιχο πεδίο του raw_std34. Στον πίνακα raw_std44 αποθηκεύονται οι μετρήσεις στάθμης για αυτή την υδρομέτρηση. Συγκεκριμένα, στο integer4 πεδίο value0 περιέχεται η μέτρηση της στάθμης που έγινε από κάποιο σταθμήμετρο του σταθμού για την υδρομέτρηση σε mm με ακρίβεια 0 ενώ στο integer4 πεδίο value1 περιέχεται ο κωδικός ΥΔΡΟΣΚΟΠΙΟΥ του οργάνου υδρομέτρησης στο οποίο αναφέρεται η στάθμη του πεδίου value0 (η ταύτιση του τύπου integer4 με τον τύπο hsiid των κωδικών ΥΔΡΟΣΚΟΠΙΟΥ είναι επιτρεπτή). Η ταυτοποίηση γίνεται από το ότι η ημερομηνία της υδρομέτρησης και η ημερομηνία της αντίστοιχης σταθμημέτρησης είναι ίδιες.

Ως παράδειγμα αναφέρεται το παρακάτω: η υδρομέτρηση έγινε στο σταθμό με κωδικό ΥΔΡΟΣΚΟΠΙΟΥ 1024 από το όργανο τύπου ΥΔΡΜΤΡ με κωδικό 1027 στις 5/12/1994

10:00. Στην πραγματική διαδικασία της υδρομέτρησης παίρνονται δύο τιμές των σταθμημέτρων, μια στην αρχή και μια στο τέλος της υδρομέτρησης. Η ημερομηνία που τελικά αποθηκεύεται για την υδρομέτρηση και για τα σταθμήμετρα είναι μια συμβατική ημερομηνία που μπορεί να είναι είτε η αρχή είτε το τέλος είτε η "μέση" της υδρομέτρησης. Τα σταθμήμετρα που χρησιμοποιήθηκαν ήταν τα 1026, 1028 και 1030. Η όλη constants (πιο συγκεκριμένα ο πίνακας constants_rp1) περιέχει χαρακτηριστικά σταθερών για διατομή ποταμού και υδραυλικά χαρακτηριστικά ποταμού:

1027, 18, ΔΙΑΤΟΜΗ, 5/12/1994 10:00, ...

1027, 20, ΥΔΡ-ΧΑΡ, 5/12/1994 10:00, ...

Ας σημειωθεί πως η ημερομηνία δημιουργίας των σταθερών δεν είναι ανάγκη να συμπίπτει με την ημερομηνία της υδρομέτρησης σε περίπτωση που για την υδρομέτρηση έχουν χρησιμοποιηθεί "παλαιότερες" σταθερές (πχ. ο ποταμός είναι εγκλιωτισμένος και δεν αλλάζουν η διατομή και τα υδραυλικά χαρακτηριστικά του).

Στους πίνακες const_sect και const_hydr είναι αποθηκευμένα τα δεδομένα των σταθερών σε μορφή:

1027, 18, ...

1027, 20, ...

όπου obj0 = 1027 (το όργανο ΥΔΡΜΤΡ στο οποίο αναφέρονται οι σταθερές) και constant = 18 ή 20.

Στον πίνακα raw_std34 αποθηκεύεται η μετρηθείσα παροχή των 5.1 m³/sec (διαίρεση με 10³) και οι σταθερές. Η λέξη κατάσταση με τιμή 1 υποδηλώνει επίπεδο 1 μόνο και είναι ενδεικτική για το συγκεκριμένο παράδειγμα:

1027, 5/12/1994 10:00, 1, 5100, 18, 20

Με τον τρόπο αυτό η προσπέλαση στις σταθερές γίνεται για τα μεν χαρακτηριστικά τους από τον constants με constants.obj0 = 1027 και constants.constant = 18 ή 20 και για τα δε στοιχεία των σταθερών από τους πίνακες const_sect για τη ΔΙΑΤΟΜΗ και const_hydr για τα ΥΔΡ-ΧΑΡ με obj0 = 1027 και constant = 18 ή 20.

Τέλος, στον πίνακα raw_std44 υπάρχουν οι μετρήσεις των σταθμημέτρων που χρησιμοποιήθηκαν στην υδρομέτρηση με μετρηθείσες τιμές στάθμης 980, 1035 και 1017 mm:

1026, 5/12/1994 10:00, 1, 980, 1027

1028, 5/12/1994 10:00, 1, 1035, 1027

1030, 5/12/1994 10:00, 1, 1017, 1027

Φαίνεται δηλαδή η ίδια ημερομηνία και το όργανο υδρομέτρησης στο οποίο αναφέρεται αυτή η σταθμημέτρηση. Επίσης πρέπει να σημειωθεί πως οι σταθμημετρήσεις αυτές καλό είναι να εισάγονται και στις "κανονικές" χρονοσειρές των συγκεκριμένων σταθμημέτρων στον πίνακα raw_std4:

1026, 5/12/1994 10:00, 1, 980

1028, 5/12/1994 10:00, 1, 1035

1030, 5/12/1994 10:00, 1, 1017

- **ΣΤΥΔΡΜΤΡ:** Η ΣΤΕΡΕΟΥΔΡΟΜΕΤΡΗΣΗ αναφέρεται σε απευθείας μέτρηση της μεταβλητής ΣΤΠΑΡΟΧΗ. Αποθηκεύεται σε μορφή std44. Στο integer4 πρώτο πεδίο value0 αποθηκεύεται η παροχή που μετρήθηκε για τη συγκεκριμένη στερεοϋδρομέτρηση σε m^3/sec με ακρίβεια 3 δεκαδικά. Στο integer4 πεδίο value1 αποθηκεύεται η στερεοπαροχή για αυτή την παροχή σε kg/sec με ακρίβεια 3 δεκαδικά. Αν στο παραπάνω παράδειγμα υδρομέτρησης είχε γίνει και στερεοϋδρομέτρηση από το ΣΤΥΔΡΜΤΡ όργανο με κωδικό ΥΔΡΟΣΚΟΠΙΟΥ 1034 τότε η εγγραφή για στερεοπαροχή $0.1 kg/sec$ θα είχε τη μορφή:

1034, 5/12/1994 10:00, 1, 5100, 100

3.9.2.20 Παροχή και στερεοπαροχή. Η παροχή και η στερεοπαροχή είναι παράγωγα όργανα. Η τιμή τους προέρχεται από τα επίσης παράγωγα όργανα ΣΤΘ-ΠΡΧ (διάγραμμα στάθμης - παροχής) και ΠΡΧ-ΣΠΡΧ (διάγραμμα παροχής - στερεοπαροχής).

- **ΠΑΡΟΧΗ, ΣΤΠΑΡΟΧΗ.** Η ΠΑΡΟΧΗ είναι η παραγόμενη τιμή της μεταβλητής ΠΑΡΟΧΗ και η ΣΤΕΡΕΟΠΑΡΟΧΗ η παραγόμενη τιμή της μεταβλητής ΣΤΠΑΡΟΧΗ. Αυτές αποθηκεύονται στο integer4 πεδίο value0 της μορφής std4 με ακρίβεια 3 δεκαδικά. Η μεν ΠΑΡΟΧΗ μετριέται σε m^3/sec η δε ΣΤΠΑΡΟΧΗ σε kg/sec .

3.9.2.21 Διαγράμματα στάθμης παροχής και παροχής - στερεοπαροχής. Το διάγραμμα στάθμης - παροχής είναι ένα παράγωγο όργανο που προέρχεται από επεξεργασία στοιχείων υδρομετρήσεων και σταθμημετρήσεων. Επειδή η "μέτρηση" του οργάνου δεν είναι απλά μία τιμή αλλά ένα διάγραμμα, αλλά και επειδή η "μέτρηση" δεν αναφέρεται σε μια συγκεκριμένη ημερομηνία αλλά σε μια ολόκληρη χρονική περίοδο (πιθανά και "για πάντα"), η αποθήκευση του οργάνου αυτού είναι ξεχωριστή και δεν ακολουθεί το γενικό κανόνα

<όργανο><ημερομηνία><χαρακτηριστικά><τιμή>

Το ίδιο ισχύει και για το διάγραμμα παροχής - στερεοπαροχής. Η μορφή αποθήκευσης του μοιάζει πολύ με αυτήν του διαγράμματος στάθμης - παροχής.

- **ΣΤΘ-ΠΡΧ.** Το ΔΙΑΓΡΑΜΜΑ ΣΤΑΘΜΗΣ ΠΑΡΟΧΗΣ αποθηκεύεται σε δύο πίνακες, ο ένας αποθηκεύει τα σημεία του διαγράμματος και ο άλλος τις καμπύλες (δεν αρκεί η αποθήκευση μόνο των καμπυλών διότι πολλά σημεία μπορεί να μην ανήκουν σε

κάποια καμπύλη). Το διάγραμμα είναι ένα παράγωγο όργανο ενός σταθμού. Αν σε έναν σταθμό γίνει άλλο διάγραμμα στάθμης - παροχής (πχ. για άλλη χρονική περίοδο) αυτό πρέπει να είναι άλλο όργανο. Γενικά πάντως συμφέρει όλες οι καμπύλες και τα σημεία στάθμης - παροχής να ανήκουν σε ένα διάγραμμα - όργανο. Κάθε διάγραμμα στάθμης - παροχής αναφέρεται σε κάποιο βασικό σταθμήμετρο που ανήκει στον ίδιο σταθμό. Ο κωδικός ΥΔΡΟΣΚΟΠΙΟΥ αυτού του σταθμήμετρου περιέχεται στο πεδίο `base_instrument` του πίνακα `instruments` για την εγγραφή του οργάνου στάθμης - παροχής. Η ημερομηνία δημιουργίας του διαγράμματος φαίνεται στο πεδίο `instruments.start_date`. Ταυτόχρονα, μια σταθερά τύπου `PAR-ORG` υπάρχει για το όργανο `ΣΤΘ-ΠΡΧ` και στον πίνακα `const_dinstr` υπάρχει η λίστα των οργάνων `ΥΔΡΜΤΡ` και `ΣΤΘΜΤΡ - ΣΤΘΓΡ` που οδήγησαν στη δημιουργία του διαγράμματος. Μετά την εισαγωγή των χαρακτηριστικών του οργάνου `ΣΤΘ-ΠΡΧ` στον πίνακα των οργάνων και των σταθερών μένει η αποθήκευση των καμπυλών του διαγράμματος στον πίνακα `raw_qh_curves` και των σημείων στον πίνακα `raw_qh_points`. Ο πίνακας `raw_qh_curves` έχει τα ακόλουθα πεδία: Ένα `hsiid` πεδίο `instrument` που περιέχει τον κωδικό ΥΔΡΟΣΚΟΠΙΟΥ του οργάνου `ΣΤΘ-ΠΡΧ`. Ένα `integer2` πεδίο `curve` που είναι ο α/α της συγκεκριμένης καμπύλης στο διάγραμμα. Έτσι μπορούν να υπάρχουν σε ένα διάγραμμα στάθμης - παροχής μέχρι 32767 καμπύλες. Δύο πεδία τύπου `hdate` με ονόματα `start_date` και `end_date` που είναι οι συγκεκριμένες ημερομηνίες αρχής και τέλους ισχύος της συγκεκριμένης καμπύλης. Σε περίπτωση που η καμπύλη ισχύει για περισσότερες από μια περιόδους πρέπει να εισαχθεί νέα εγγραφή στον πίνακα `raw_qh_curves`. Δυο πεδία τύπου `char(25)` με ονόματα `start_period` και `end_period` μπορούν να πάρουν τιμές τις περιόδους αρχής και τέλους ισχύος της συγκεκριμένης καμπύλης, πχ. `ΑΝΟΙΕΗ,` `ΧΕΙΜΩΝΑΣ`. Ένα πεδίο `integer1`, το `log` παίρνει τιμή διάφορη του 0 αν η καμπύλη είναι λογαριθμική και 0 αν δεν είναι. Ένα ακόμα πεδίο `integer1`, το `ext` παίρνει τιμή διάφορη του 0 αν η καμπύλη προέρχεται από επέκταση και 0 αν όχι. Ένα πεδίο `integer4` με όνομα `h_offset` είναι η αρχική μετατόπιση { Γ } της στάθμης σε mm με ακρίβεια 0 δεκαδικά. Τέλος ένα πεδίο `char(40)` το `remarks` περιέχει διάφορες παρατηρήσεις για τη συγκεκριμένη καμπύλη. Στον πίνακα `raw_qh_points` αποθηκεύονται τα σημεία του διαγράμματος. Έχει τα πεδία: `instrument`, τύπου `hsiid` είναι βέβαια ο κωδικός ΥΔΡΟΣΚΟΠΙΟΥ του οργάνου `ΣΤΘ-ΠΡΧ`, `curve` τύπου `integer2` είναι ο α/α της καμπύλης στην οποία ανήκει το σημείο και αντιστοιχεί στο πεδίο `curve` του πίνακα `raw_qh_curves`, έχει δε τιμή 0 αν το σημείο δεν ανήκει σε καμία καμπύλη. Το πεδίο `flag`, τύπου `integer1` περιέχει τις ιδιότητες του σημείου. Τα `integer4` πεδία `q` και `h` είναι οι τιμές της παροχής σε m^3/sec με ακρίβεια 3 δεκαδικά και της στάθμης σε mm με ακρίβεια 0 δεκαδικά που αποτελούν τις συντεταγμένες του

σημείου πάνω στο διάγραμμα. Με αυτό τον τρόπο, ένα διάγραμμα στάθμης - παροχής δημιουργείται αν σχεδιασθούν πρώτα όλα τα σημεία του πίνακα `raw_gh_points` για το συγκεκριμένο όργανο και στη συνέχεια αν κάθε καμπύλη του πίνακα `raw_gh_curves` σχεδιασθεί ενώνοντας όλα τα σημεία που ανήκουν σε αυτή.

Ως παράδειγμα των παραπάνω, έστω το όργανο 1040 που είναι τύπου ΣΤΘ-ΠΡΧ και έχει ως βασικό όργανο το ΣΤΘΜΤΡ 1026 (`instruments.base_instrument = 1026`), ενώ για τη δημιουργία του χρησιμοποιήθηκαν στοιχεία και από τα ΣΤΘΜΤΡ 1028 και 1030. Τότε θα υπάρχει εγγραφή στον πίνακα `constants`:

```
1040, 1, ΠΑΡ-ΟΡΓ, 25/12/1994 02:00, ...
```

και σειρά εγγραφών στον πίνακα `const_dinstr`:

```
1040, 1, 1026
```

```
1040, 1, 1028
```

```
1040, 1, 1030
```

```
1040, 1, 1027
```

Αν το διάγραμμα στάθμης παροχής έχει τη μορφή του παρακάτω σχήματος τότε στον πίνακα `raw_gh_curves` θα υπάρχουν οι εγγραφές:

```
1040, 1, 1/6/1993 00:00, 29/6/1993 00:00, null, null, 0, 0, 0
```

```
1040, 2, 1/1/1 00:00, 1/1/1 00:00, ΧΕΙΜΩΝΑΣ, ΑΝΟΙΞΗ, 0, 0, 0
```

```
1040, 2, 1/10/1993 00:00, 31/10/1993 00:00, null, null, 0,0,0
```

```
1040, 3, 30/6/1993 00:00, 30/9/1993 00:00, null, null, 0,0,9
```

```
1040, 3, 1/11/1993 00:00, 30/11/1993 00:00, null, null, 0,0,9
```

```
1040, 4, 1/1/1 00:00, 1/1/1 00:00, ΧΕΙΜΩΝΑΣ, ΑΝΟΙΞΗ, 0,1,0
```

```
1040, 4, 1/10/1993 00:00, 31/10/1993 00:00, null, null, 0,1,0
```

Στον πίνακα `raw_gh_points` θα υπάρχουν οι ακόλουθες εγγραφές:

```
1040, 1, 0, 600, 1150
```

```
1040, 0, 0, 810, 1350
```

```
1040, 1, 0, 800, 1550
```

```
1040, 1, 0, 1500, 1800
```

```
1040, 0, 0, 1900, 2000
```

```
1040, 1, 0, 2000, 1950
```

```
1040, 1, 0, 3000, 2000
```

```
1040, 2, 0, 1400, 1160
```

```
1040, 2, 0, 1400, 1400
```

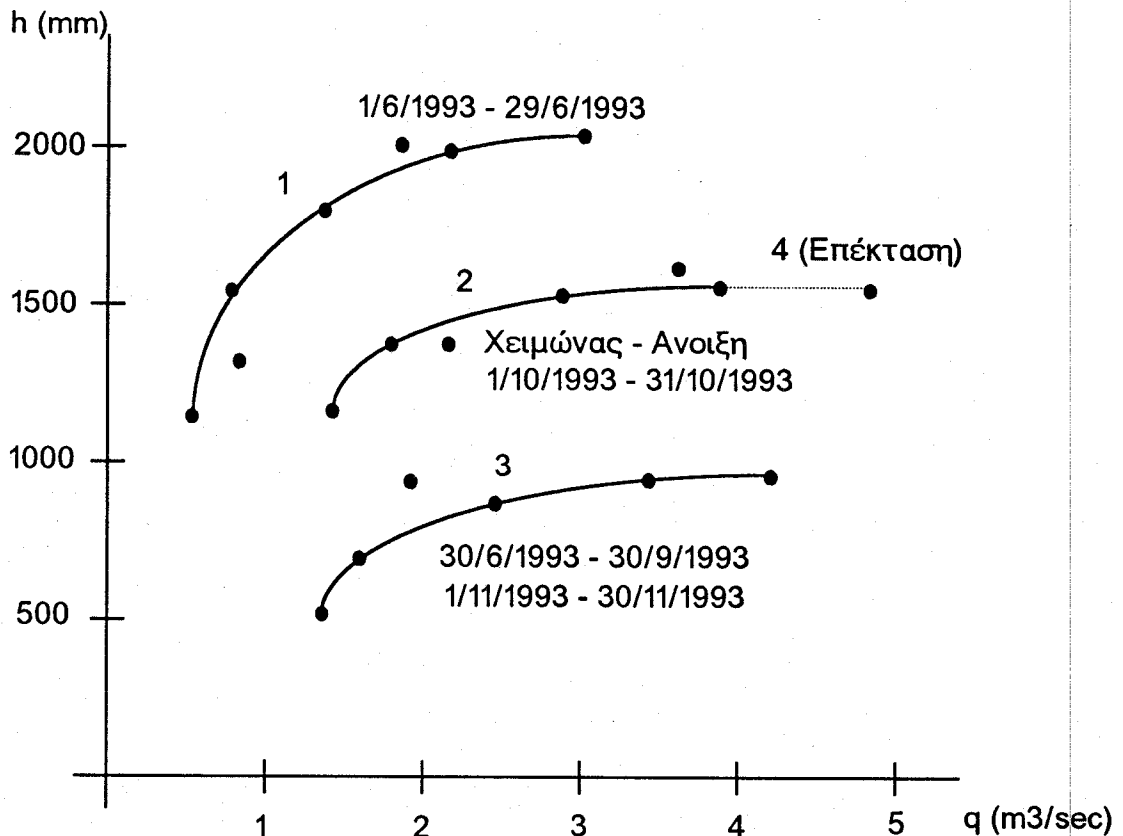
```
1040, 0, 0, 2000, 1380
```

```
1040, 2, 0, 2800, 1500
```

```
1040, 0, 0, 3700, 1550
```

```
1040, 2, 0, 3900, 1500
```

1040, 4, 0, 4800, 1500
 1040, 3, 0, 1300, 550
 1040, 3, 0, 1600, 700
 1040, 0, 0, 1900, 900
 1040, 3, 0, 2400, 700
 1040, 3, 0, 3400, 900
 1040, 3, 0, 4250, 950



Σχήμα 3.9.1. Σκαρίφημα διαγράμματος στάθμης - παροχής

- ΠΡΧ-ΣΠΡΧ. Το ΔΙΑΓΡΑΜΜΑ ΠΑΡΟΧΗΣ ΣΤΕΡΕΟΠΑΡΟΧΗΣ επίσης αποθηκεύεται σε δύο πίνακες, τον raw_qs_q_curves που αναφέρεται στις καμπύλες του διαγράμματος και τον raw_qs_q_points που αναφέρεται στα σημεία του. Ο πίνακας raw_qs_q_points είναι ίδιος στη μορφή και τη λειτουργικότητα του με τον πίνακα raw_qh_points του διαγράμματος στάθμης - παροχής εκτός του πεδίου h που έχει αντικατασταθεί με το πεδίο qs που είναι τύπου integer4 και αποθηκεύει στερεοπαροχή σε kgf/sec με ακρίβεια 3 δεκαδικών. Έτσι οι συντεταγμένες κάθε σημείου του διαγράμματος παροχής - στερεοπαροχής είναι οι q και qs αντί των q και

h του διαγράμματος στάθμης - παροχής. Αντίστοιχα ο πίνακας `raw_qsqr_curves` έχει κοινά σε τύπο και λειτουργικότητα πεδία `instrument`, `curve`, `start_date`, `end_date` και `remarks` με τον `raw_qh_curves` του διαγράμματος στάθμης - παροχής. Τα πεδία `log`, `ext` και `h_offset` δεν υπάρχουν. Τέλος τα πεδία `start_period` και `end_period` είναι τύπου `integer1` και αντί για λεκτική περιγραφή περιέχουν τον αριθμό του μήνα που το διάγραμμα αρχίζει και παύει αντίστοιχα να ισχύει.

3.9.2.22 Πλοία. Διάφορα πλοία που έπειτα από κατάλληλη συμφωνία μεταδίδουν διάφορες μετεωρολογικές πληροφορίες στην ΕΜΥ μπορούν να θεωρηθούν ως "κινητοί" μετεωρολογικοί σταθμοί. Με το "κινητοί" εννοείται πως δεν έχουν φ και λ. Τα πλοία μπορούν να εισάγονται στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ ως δευτερεύοντες μετεωρολογικοί σταθμοί που περιέχουν διάφορα όργανα. Προς το παρόν τα όργανα των πλοίων που δεν είναι κοινά με αντίστοιχα όργανα ξηράς (πχ. ΘΡΜΜΤΡ, ΒΡΧΜΤΡ) δεν περιλαμβάνονται στη ΒΔ, ωστόσο είναι πολύ εύκολο να εισαχθούν, όπως και οποιοδήποτε νέο όργανο.

3.9.3 Εισαγωγή νέας μεταβλητής και οργάνου. Η εισαγωγή νέας μεταβλητής στη ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ είναι πολύ απλή: αρκεί να εισαχθεί η αντίστοιχη εγγραφή στον πίνακα `variables`. Επίσης πρέπει να ειδοποιηθούν και οι διαχειριστές των άλλων κόμβων ώστε να ενημερώσουν κατάλληλα τις δικές τους ΤΒΔ. Η εισαγωγή νέου οργάνου είναι επίσης σχετικά απλή υπόθεση. Καταρχήν αποφασίζεται το (μοναδικό) όνομα και η μορφή ή οι μορφές αποθήκευσης του νέου οργάνου. Συνιστάται σαφώς να χρησιμοποιηθεί κάποιος από τους υπάρχοντες τρόπους αποθήκευσης, ακόμα και αν αυτό σημαίνει κάποια σχετική "σπατάλη" χώρου (πχ. χρήση μορφών `1st`) γιατί σε διαφορετική περίπτωση πρέπει να δημιουργηθούν νέοι πίνακες `raw` και `aggr` όχι μόνο στην ΤΒΔ αλλά και στις ΚΒΔ, διαδικασία επίπονη και σύνθετη. Αφού λοιπόν βρεθούν τα στοιχεία αυτά, εισάγεται κατάλληλη εγγραφή στον πίνακα `instrument_ts`. Στη συνέχεια εισάγεται η περιγραφή των πεδίων κάθε μορφής αποθήκευσης που μπορεί να χρησιμοποιηθεί με την ακρίβεια σε δεκαδικά ψηφία και τις μονάδες μέτρησης για κάθε πεδίο της μορφής στον πίνακα `precision_unit`. Ο α/α του πεδίου, δηλαδή το επίθεμα `{ Γ } <n>` του ονόματος `value<n>` του πεδίου της μορφής αποθήκευσης είναι το πεδίο `seq` του πίνακα `precision_unit`. Αν το `seq` έχει τιμή -1 τότε αυτό σημαίνει πως η ίδια ακρίβεια και μονάδα μέτρησης ισχύουν για όλα τα πεδία. Για περισσότερες λεπτομέρειες βλ. παρακάτω.

Μετά βέβαια την εισαγωγή των οργάνων πρέπει να αναπτυχθούν και οι κατάλληλες εφαρμογές εισαγωγής και διαχείρισης των χρονοσειρών τους.

4 ΥΠΟΣΥΣΤΗΜΑΤΑ ΑΣΦΑΛΕΙΑΣ ΚΑΙ ΧΡΕΩΣΗΣ

4.1 Γενικά για το υποσύστημα ασφάλειας

Το υποσύστημα ασφάλειας της ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ βασίζεται στην ιδέα των ομάδων χρηστών { Γ } [Πιπ92β] και λειτουργεί με βάση τα ακόλουθα κύρια σημεία:

- Κάθε χρήστης προσπελαίνει το σύστημα με βάση έναν κωδικό χρήστη { Γ } που του χορηγείται από το διαχειριστή συστήματος { Γ }.
- Για να μπορέσει αυτός ο κωδικός χρήστη να χρησιμοποιήσει τις υπηρεσίες του ΣΔΒΔ INGRES που χρησιμοποιείται πρέπει να εισαχθεί καταρχήν στους νόμιμους χρήστες του από το διαχειριστή της INGRES { Γ } [INGRES91d][INGRES91h].
- Για να μπορέσει ο κωδικός χρήστη να χρησιμοποιήσει τις υπηρεσίες του ΥΔΡΟΣΚΟΠΙΟΥ πρέπει να είναι καταρχήν νόμιμος χρήστης της INGRES και στη συνέχεια να εισαχθεί στους νόμιμους χρήστες του ΥΔΡΟΣΚΟΠΙΟΥ από το διαχειριστή ΒΔ { Γ }.
- Αναπόσπαστη παράμετρος κάθε σύνδεσης με τη ΒΔ είναι ο κωδικός χρήστη που την εκτελεί. Οι κωδικοί με τους οποίους γίνεται η σύνδεση είναι γενικά αυτοί με τους οποίους γίνεται η προσπέλαση στο σύστημα.
- Καθορίζονται ομάδες χρηστών. Κάθε ομάδα αποτελείται από έναν ή περισσότερους χρήστες ή, ορθότερα, κωδικούς χρηστών. Κάθε κωδικός χρήστη μπορεί να ανήκει σε περισσότερες από μία ομάδες.
- Πάνω σε συγκεκριμένα αντικείμενα μιας ΒΔ, στο σύνολο μιας ΒΔ αλλά και στο σύνολο της εγκατάστασης της INGRES { Γ } μπορούν να ορισθούν διάφορα δικαιώματα - προνόμια πρόσβασης ή, αλλιώς, δυνατές ενέργειες. Επίσης μπορεί να καθορισθεί ποιά από αυτά τα δικαιώματα είναι διαθέσιμα και σε ποιούς κωδικούς ή ποιές ομάδες. Καθορίζεται δηλαδή από τον κατάλληλο διαχειριστή μια τριάδα της μορφής
(αντικείμενα / ΒΔ / εγκατάσταση,
δικαίωμα - προνόμιο πρόσβασης,
κωδικός χρήστη / ομάδα)
που σημαίνει πως ο κωδικός χρήστη ή η ομάδα έχουν το αναφερόμενο δικαίωμα - προνόμιο πρόσβασης στα αντικείμενα ή τη ΒΔ ή το σύνολο της εγκατάστασης.
- Η προσπέλαση στο σύστημα και η χρήση του ΣΔΒΔ INGRES με κάποιο συγκεκριμένο κωδικό χρήστη του δίνει τα δικαιώματα - προνόμια πρόσβασης που συνδέονται με αυτό το χρήστη για το σύνολο της εγκατάστασης. Η προσπέλαση στη

ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ με τον ίδιο κωδικό χρήστη δίνει τα δικαιώματα - προνόμια πρόσβασης που συνδέονται με αυτό τον κωδικό για τη συγκεκριμένη ΒΔ. Τέλος η πρόσβαση στα διάφορα αντικείμενα της ΒΔ δίνει τα δικαιώματα - προνόμια που αναφέρονται στον κωδικό χρήστη για αυτά τα αντικείμενα.

- Οι ομάδες δεν έχουν προνόμια πάνω στην εγκατάσταση, έχουν όμως πάνω στο σύνολο της ΒΔ και σε επιμέρους αντικείμενα της. Έτσι, κάθε σύνδεση με κάποια ΒΔ έχει ως παράμετρο, εκτός του κωδικού χρήστη, και κάποιο κωδικό ομάδας. Ο χρήστης δεν μπορεί γενικά να καθορίσει τον κωδικό χρήστη που θα χρησιμοποιήσει, μπορεί όμως να καθορίσει την ομάδα, αρκεί βέβαια να ανήκει σε αυτήν. Μια σύνδεση ως μέλος μιας ομάδας δίνει στο χρήστη τα δικαιώματα - προνόμια της ομάδας αυτής, πλέον αυτών που έχει ως χρήστης.
- Στο σύστημα του ΥΔΡΟΣΚΟΠΙΟΥ γενικά δεν καθορίζονται προνόμια για συγκεκριμένους χρήστες αλλά για ομάδες χρηστών. Κάθε ομάδα έχει μια συγκεκριμένη αποστολή που της δίνει δικαιώματα πρόσβασης σε αντικείμενα της ΒΔ και στο σύνολο της ΒΔ. Κάθε χρήστης που θέλει να εκτελέσει εργασίες σχετιζόμενες με αυτή την αποστολή συνδέεται με τη ΒΔ με παράμετρο σύνδεσης την αντίστοιχη ομάδα.
- Η πρόσβαση στους απομακρυσμένους κόμβους δεν γίνεται με βάση τους κωδικούς χρήστη και τις ομάδες στον τοπικό κόμβο. Αντίθετα, όλες οι προσβάσεις εκτός του τοπικού κόμβου αντιστοιχίζονται σε έναν κωδικό για το συγκεκριμένο Φορέα, με χρήση των μηχανισμών του υποσυστήματος δικτύωσης της INGRES, INGRES/NET [INGRES91c] και σε μια ομάδα "απομακρυσμένων χρηστών". Δηλαδή, ακόμα και αν ένας κωδικός *abcd* ανήκει σε μια "παντοδύναμη" ομάδα *efgh* στον κόμβο Α, για πρόσβαση εκτός του Α ο κωδικός αυτός αντιστοιχίζεται σε κάποιον άλλον με περιορισμένα προνόμια.

4.2 Κατηγορίες αντικειμένων και δικαιωμάτων πρόσβασης

Στην τριάδα (αντικείμενα / ΒΔ / εγκατάσταση, δικαίωμα - προνόμιο πρόσβασης, κωδικός χρήστη / ομάδα) ο συνδυασμός του πρώτου και του δεύτερου μέλους μπορεί να είναι:

4.2.1 Αντικείμενα: πίνακες. Οι πίνακες και οι όψεις μιας συγκεκριμένης ΒΔ είναι μια μεγάλη κατηγορία αντικειμένων. Πάνω σε αυτά τα αντικείμενα, όλα μαζί ή μεμονωμένα, οι δυνατές ενέργειες - δικαιώματα πρόσβασης είναι:

- **select.** Το δικαίωμα δηλαδή της ανάκτησης πληροφοριών από το αντικείμενο.
- **insert.** Το δικαίωμα δηλαδή της εισαγωγής νέων πληροφοριών στο αντικείμενο

- **update & delete.** Το δικαίωμα δηλαδή της τροποποίησης και διαγραφής πληροφοριών από το αντικείμενο. Τα δύο αυτά δικαιώματα είναι ξεχωριστά αλλά στα πλαίσια της ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ θεωρείται πως όποιος έχει το ένα λογικό είναι να έχει και το άλλο.
- **all.** Συνδυασμός όλων των παραπάνω δικαιωμάτων.

Οι πίνακες και οι όψεις της ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ χωρίζονται επιπλέον σε υποκατηγορίες:

- **Πίνακες data.** Πρόκειται για τους πίνακες που αποθηκεύουν τα δεδομένα των χρονοσειρών και των σταθερών.
- **Πίνακες admin και admin-rmt.** Οι πίνακες admin είναι αυτοί που αποθηκεύουν τις διαχειριστικές πληροφορίες για σταθμούς, όργανα, χαρακτηριστικά σταθερών και χρονοσειρών και γεγονότα. Υποκατηγορία των πινάκων admin είναι οι πίνακες admin-rmt οι οποίοι είναι οι διαχειριστικοί πίνακες της κατηγορίας rmt, αυτοί δηλαδή που μπορούν να εγγραφούν από κατάλληλους δαίμονες απομακρυσμένων κόμβων.
- **Άλλοι πίνακες.** Πρόκειται για τους υπόλοιπους πίνακες της ΒΔ, κύρια δηλαδή αυτούς των πληροφοριών εφαρμογών.

4.2.2 Αντικείμενα: διαδικασίες ΒΔ. Οι διαδικασίες ΒΔ είναι η δεύτερη μεγάλη κατηγορία αντικειμένων. Πάνω στις διαδικασίες ΒΔ είναι δυνατή μια μόνη ενέργεια:

- **execute.** Το δικαίωμα δηλαδή εκτέλεσης της διαδικασίας ΒΔ. Αυτό το δικαίωμα έχει και ειδική σημασία: αν μια διαδικασία ΒΔ προσπελαύνει κάποιους πίνακες με κάποιο συγκεκριμένο τρόπο πχ. για εισαγωγή στοιχείων και σε κάποια ομάδα χρηστών ή σε μεμονωμένους χρήστες δοθεί δικαίωμα εκτέλεσης της διαδικασίας, τότε οι ομάδες ή οι χρήστες μπορούν να προσπελάσουν τους πίνακες με τον ίδιο συγκεκριμένο τρόπο χωρίς απαραίτητα να έχουν το αντίστοιχο δικαίωμα πρόσβασης πάνω στους πίνακες. Με αυτό τον τρόπο ελέγχεται καλύτερα η πρόσβαση στους πίνακες. Το χαρακτηριστικό αυτό χρησιμοποιείται για την εισαγωγή, τροποποίηση και διαγραφή πληροφοριών σε διαχειριστικούς πίνακες admin: οι ομάδες χρηστών που είναι επιφορτισμένες με τα καθήκοντα αυτά δεν έχουν τα αντίστοιχα δικαιώματα πρόσβασης insert, update & delete. Έχουν όμως δικαίωμα execute σε διαδικασίες ΒΔ που με τη σειρά τους εκτελούν ελεγχόμενα αυτές τις ενέργειες.

4.2.3 Ολόκληρη η ΒΔ. Ορισμένα δικαιώματα αναφέρονται στο σύνολο της ΒΔ [INGRES91d], στην περίπτωση του ΥΔΡΟΣΚΟΠΙΟΥ της db0. Τα σημαντικότερα από αυτά είναι:

- **create_table.** Το δικαίωμα δηλαδή δημιουργίας νέου πίνακα στη ΒΔ.

- **create_procedure.** Το δικαίωμα δηλαδή δημιουργίας νέας διαδικασίας στη ΒΔ.
- **query_row_limit.** Ο μέγιστος αριθμός γραμμών - εγγραφών που μπορεί να επιστρέψει ως απάντηση μια ερώτηση προς τη ΒΔ. Αν ο προβλεπόμενος από το βελτιστοποιητή ερωτήσεων { Γ } αριθμός γραμμών απάντησης είναι μεγαλύτερος τότε η ερώτηση δεν εκτελείται καθόλου. Με τον τρόπο αυτό είναι δυνατό να αποφευχθεί η υπερβολική χρήση της ΒΔ πάνω από το δίκτυο. Για παράδειγμα απορρίπτονται ερωτήσεις του είδους: "Άς φέρουμε όλες τις πληροφορίες και μετά βλέπουμε τι ακριβώς χρειαζόμαστε".

Υπάρχουν και άλλα δικαιώματα τα οποία δεν χρησιμοποιούνται και έχουν γενικά τις εξ' ορισμού τιμές (συνήθως "ΟΧΙ").

4.2.4 Σύνολο της εγκατάστασης. Υπάρχουν τέλος δικαιώματα που αναφέρονται στη συνολική πρόσβαση του συγκεκριμένου ΣΔΒΔ. Τα σημαντικότερα από αυτά είναι:

- **create_data_base.** Το δικαίωμα δηλαδή της δημιουργίας νέων ΒΔ.
- **super_user.** Γενικά οι χρήστες δεν έχουν δικαίωμα χρήσης άλλου κωδικού χρήστη. Σε περίπτωση που ένας χρήστης διαθέτει περισσότερους από έναν κωδικούς και θέλει να χρησιμοποιήσει άλλον από τον παρόντα πρέπει να αποσυνδεθεί { Γ } από το σύστημα και να επανασυνδεθεί { Γ } με το νέο κωδικό. Αν όμως έχει το προνόμιο **super_user** έχει δικαίωμα να "υποδυθεί" τον άλλο κωδικό και να συνδεθεί με τη ΒΔ με αυτόν, σαν να είχε κανονικά προσπελάσει το σύστημα με αυτόν. Αυτό γίνεται με χρήση της παραμέτρου -u, όπως στο παράδειγμα: `sql -uotheruser db0`.

4.3 Κατηγορίες ομάδων χρηστών

Κάποιος χρήστης που συνδέεται με τη ΒΔ αποκτά γενικά τα δικαιώματα της εξ' ορισμού ομάδας του, εκτός και αν ζητήσει σαφώς τη σύνδεση κάτω από άλλη ομάδα, αν βέβαια ανήκει σε αυτή. Οι ομάδες έχουν γενικά δικαιώματα `select` σε όλους τους πίνακες, εκτός ορισμένων εξαιρέσεων. Επίσης όλοι οι χρήστες έχουν δικαίωμα χρήσης των διαδικασιών ΒΔ των σχετικών με τη χρέωση (βλ. παρακάτω). Τέλος, όλοι οι χρήστες έχουν δικαίωμα δημιουργίας πινάκων στη ΒΔ (`create_table` προνόμιο). Το δικαίωμα αυτό είναι αναγκαίο (δυστυχώς) για τη λειτουργία της ΚΒΔ και δεν μπορεί να απενεργοποιηθεί. Ο διαχειριστής της ΒΔ θα πρέπει να ελέγχει περιοδικά τη ΒΔ για την παρουσία "άσχετων" πινάκων και να τους διαγράφει. Οι διαθέσιμες ομάδες χρηστών ακολουθούν:

- **public.** Είναι το κοινό, το σύνολο δηλαδή των χρηστών του συστήματος. Το δικαίωμα `query_row_limit` έχει τιμή 10000, δηλαδή εκτελούνται ερωτήσεις με προβλεπόμενο μήκος απάντησης μέχρι 10000 γραμμές - εγγραφές. Οι "μεγαλύτερες"

ερωτήσεις απορρίπτονται χωρίς να εκτελεσθούν. Το όριο αυτό μπορεί να αναπροσαρμοστεί μετά από μελέτη της κυκλοφορίας στο δίκτυο.

- **husers.** Πρόκειται για την εξ' ορισμού ομάδα για τους τοπικούς χρήστες, αυτούς δηλαδή που βρίσκονται στον ίδιο κόμβο. Κάθε τοπικός χρήστης ανήκει τουλάχιστον στην ομάδα αυτή. Οριο `query_row_limit` 50000.
- **hrusers.** Πρόκειται για την εξ' ορισμού ομάδα για τους απομακρυσμένους χρήστες, αυτούς δηλαδή που βρίσκονται σε όλους τους άλλους κόμβους. Κάθε εξωτερική πρόσβαση γίνεται κανονικά κάτω από αυτή την ομάδα. Οριο `query_row_limit` 10000 γραμμές.
- **hdatainsert.** Οποιος χρήστης ανήκει σε αυτή την ομάδα έχει το δικαίωμα να εισάγει δεδομένα στους πίνακες `data`. Έχει δηλαδή δικαιώματα `insert` σε αυτούς. Κανένα όριο `query_row_limit`. Προορισμός αυτής της ομάδας είναι η εισαγωγή δεδομένων χρονοσειρών.
- **hdataupdate.** Οποιος χρήστης ανήκει σε αυτή την ομάδα έχει το δικαίωμα να τροποποιεί και να διαγράφει δεδομένα των πινάκων `data`. Έχει δηλαδή δικαιώματα `update` & `delete` σε αυτούς. Κανένα όριο `query_row_limit`. Τα μέλη της ομάδας αυτής μπορούν να ελέγχουν και να τροποποιήσουν τα δεδομένα χρονοσειρών που έχουν ήδη εισαχθεί.
- **hadmininsert.** Οποιος χρήστης ανήκει σε αυτή την ομάδα έχει το δικαίωμα να εισάγει νέες διαχειριστικές πληροφορίες στους πίνακες `admin`. Η εισαγωγή αυτή δεν γίνεται απευθείας (δηλαδή η ομάδα δεν έχει δικαίωμα `insert` στους πίνακες) αλλά μέσω δικαιώματος `execute` των κατάλληλων διαδικασιών `BD_station()`, `_instrument()`, `_timeseries()`, `_constant()` και `_event()` για εισαγωγή στους πίνακες σταθμών, οργάνων, χαρακτηριστικών χρονοσειρών και σταθερών και γεγονότων. Το ό,τι οι διαδικασίες μπορούν να τροποποιήσουν τους πίνακες οφείλεται στο γεγονός ότι τις έχει δημιουργήσει ο ιδιοκτήτης των πινάκων και διαχειριστής της ΒΔ (`hydro`). Κανένα όριο `query_row_limit`. Προορισμός αυτής της ομάδας είναι η εισαγωγή διαχειριστικών πληροφοριών.
- **hadminupdate.** Οποιος χρήστης ανήκει σε αυτή την ομάδα έχει δικαίωμα να τροποποιήσει και να διαγράψει διαχειριστικές πληροφορίες των πινάκων `admin`. Και σε αυτή την περίπτωση η τροποποίηση - διαγραφή δεν γίνεται απευθείας (δηλαδή η ομάδα δεν έχει δικαιώματα `update` - `delete`) αλλά μέσω των διαδικασιών ΒΔ. Κανένα όριο `query_row_limit`. Οποιος ανήκει σε αυτή την ομάδα έχει το δικαίωμα ελέγχου και τροποποίησης των διαχειριστικών πληροφοριών.
- **hrpl.** Οποιος χρήστης ανήκει σε αυτή την ομάδα έχει δικαίωμα εισαγωγής, τροποποίησης και διαγραφής πληροφοριών των διαχειριστικών πληροφοριών των

πινάκων `admin_rmt` δηλαδή των τύπου `rmt`. Ας σημειωθεί επίσης πως η ομάδα αυτή δεν έχει κανένα δικαίωμα `insert`, `update` & `delete` στους υπόλοιπους πίνακες `admin`, όπως και οι ομάδες `hadmininsert` και `hadminupdate` δεν έχουν δικαιώματα `insert`, `update` & `delete` στους πίνακες `admin_rmt`. Η ομάδα αυτή είναι η εξ' ορισμού ομάδα για την πρόσβαση των απομακρυσμένων δαιμόνων αντιγραφής `hrpld`. Κανένα όριο `query_row_limit`.

- **hdba.** Η ομάδα αυτή είναι η ομάδα των διαχειριστών της ΒΔ. Έχει όλα τα δικαιώματα σε όλα τα αντικείμενα της ΒΔ. Είναι προφανές πως η ένταξη κάποιου χρήστη σε αυτή την ομάδα του δίνει απεριόριστα προνόμια και πρέπει να γίνεται με εξαιρετική προσοχή. Κανένα όριο `query_row_limit`.
- **hdev.** Η ομάδα αυτή είναι η ομάδα των χρηστών που αναπτύσσουν λογισμικό για το ΥΔΡΟΣΚΟΠΙΟ. Έχουν επίσης απεριόριστα δικαιώματα, σαν την ομάδα `hdba` για να διευκολύνονται στην ανάπτυξη του λογισμικού. Η παρουσία όμως ενός χρήστη σε αυτή την ομάδα, ανεξάρτητα από την απαιτούμενη προσοχή στην ένταξη του, πρέπει να είναι προσωρινή, για το διάστημα ανάπτυξης του λογισμικού. Κανένα όριο `query_row_limit`.

Κάτι που πρέπει σε κάθε περίπτωση να γίνει σαφές είναι το ό,τι σε οποιαδήποτε ομάδα και αν ανήκει ένας χρήστης στο τοπικό σύστημα, σε όλα τα απομακρυσμένα συστήματα αυτός ανήκει στην ομάδα `hrusers` που έχει περιορισμένα δικαιώματα, εκτός και αν ο διαχειριστής ΒΔ κάποιου απομακρυσμένου κόμβου, μετά από κατάλληλη συμφωνία, του δώσει περισσότερα προνόμια εντάσσοντας τον σε κάποια "ισχυρότερη" ομάδα.

Ο παρακάτω πίνακας συνογίζει τον τρόπο πρόσβασης των διαφόρων ομάδων στις διάφορες κατηγορίες αντικειμένων:

Είδος	data	admin	admin-rmt	Άλλα	Διαδικασίες	Όριο γραμμών
Ομάδα						
public	S	S	S	S	EA	10000
husers	S	S	S	S	EA	50000
hrusers	S	S	S	S	EA	10000
hdatainsert	S, I	S	S	S	EA	N
hdataupdate	S, I, U, D	S	S	S	EA	N

hadmininsert	S	S και I μέσω διαδικα- σίας	S	S	EA, ΕΔ	N
hadminupdate	S	S και I, U, D μέσω διαδικα- σίας	S	S	EA, ΕΔ	N
hrpl	S	S	S, I, U, D	S	EA, ΕΔ	10000
hdba	A	A	A	A	A	N
hdev	A	A	A	A	A	N

Πίνακας 4.3.1: Οι δυνατές ενέργειες ανά ομάδα και κατηγορία πίνακα. S = Select, I = insert, U = update, D = delete, A = all, EA = Execute για διαδικασίες χρέωσης, ΕΔ = execute για διαδικασίες διαχειριστικών πληροφοριών, N = None (Κανένα).

Σε σχέση με την έννοια της τριάδας (αντικείμενα / ΒΔ / εγκατάσταση, δικαίωμα - προνόμιο πρόσβασης, κωδικός χρήστη / ομάδα) ορισμένες χαρακτηριστικές τριάδες είναι οι ακόλουθες:

- (raw_std2, select, public). Δηλαδή το κοινό public έχει δικαίωμα ανάκτησης πληροφοριών select από τον πίνακα raw_std2.
- (raw_std2, insert & update & delete, hdataupdate). Η ομάδα hdataupdate έχει δικαίωμα τροποποίησης και διαγραφής πληροφοριών από τον πίνακα raw_std2.
- (stations_rpl, select, public) και (_station(), execute, hdatainsert). Το κοινό public έχει δικαίωμα επιλογής από τον πίνακα stations_rpl αλλά μόνο η ομάδα hdatainsert έχει δικαίωμα εκτέλεσης execute πάνω στη διαδικασία ΒΔ _station() η οποία είναι αυτή που τελικά εισάγει στοιχεία στη ΒΔ.
- (db0, query_row_limit = 10000, hrusers). Η ομάδα hrusers έχει όριο εγγραφών query_row_limit τις 10000 πάνω στη ΒΔ db0.

Οι τριάδες καθορίζονται με εντολές SQL grant όπως οι ακόλουθες που υλοποιούν το πρώτο και το δεύτερο από τα παραπάνω παραδείγματα:

```
grant select on raw_std2 to public;
grant insert, update, delete on raw_std2 to group hdataupdate;
```

Η δεύτερη εντολή σε συνδυασμό με την πρώτη δίνουν για την ομάδα hdataupdate:

```
grant all on raw_std2 to group hdataupdate;
```

Κάποιος κωδικός χρήστη μπορεί να έχει πολλαπλά δικαιώματα πρόσβασης σε κάποιο αντικείμενο μια και μπορεί να καθορισθούν με κατάλληλες εντολές grant δικαιώματα για το κοινό, για την ομάδα στην οποία ανήκει ο κωδικός χρήστη και μεμονωμένα για τον κωδικό. Σε αυτή την περίπτωση ο κωδικός έχει "τελικά" δικαιώματα χρήστη τη λογική ένωση των δικαιωμάτων των τριών κατηγοριών. Αυτό φαίνεται και στο παραπάνω παράδειγμα, όπου οι κωδικοί που ανήκουν στην ομάδα hdataupdate έχουν όλα τα δικαιώματα πρόσβασης μια και το select του κοινού public "ένώνεται" με τα insert, update, delete της ομάδας.

Σε περίπτωση τώρα που ο διαχειριστής ΒΔ σε κάποιο κόμβο επιθυμεί διαφορετική πολιτική (πχ. θέλει να αφαιρέσει κάποια δικαιώματα πρόσβασης από κάποιους πίνακες) αρκεί να χρησιμοποιήσει κατάλληλες εντολές grant και revoke.

4.4 Κωδικοί χρήστη

Σε κάθε κόμβο του ΥΔΡΟΣΚΟΠΙΟΥ πρέπει να υπάρχουν οι ακόλουθοι εξ' ορισμού κωδικοί χρήστη:

- **Διαχειριστής ΒΔ hydro.** Ο χρήστης αυτός είναι ο ιδιοκτήτης και διαχειριστής της ΤΒΔ του κόμβου, των προγραμμάτων εφαρμογών και των αρχείων του ΥΔΡΟΣΚΟΠΙΟΥ. Έχει πλήρη δικαιώματα πρόσβασης σε αυτά τα αντικείμενα. Τα δικαιώματα αυτά είναι ισχυρότερα απ' όσα δίνει η συμμετοχή στην ομάδα hdba, μια και επιτρέπεται τροποποίηση ή και διαγραφή των αντικειμένων και όχι μόνο των περιεχομένων τους. Επίσης ο χρήστης hydro έχει και δικαίωμα σύνδεσης με τη ΒΔ ως άλλος χρήστης με χρήση της παραμέτρου -u όπως πχ. `sql -usomeuser db0`, έχει δηλαδή προνόμιο `super_user`. Αυτό το δικαίωμα δεν υπάρχει για τους συνήθεις χρήστες.
- **Διαχειριστής INGRES ingres.** Ο χρήστης αυτός είναι ο διαχειριστής του ΣΔΒΔ INGRES και είναι σε θέση να προσδέσει νέους χρήστες της INGRES, νέες πληροφορίες για το υποσύστημα δικτύωσης INGRES/NET και τα άλλα υποσυστήματα και παραμέτρους της INGRES κοκ. Επίσης, σε αυτόν ανήκουν τα αρχεία και προγράμματα της INGRES. Έχει και αυτός το δικαίωμα να χρησιμοποιήσει την παράμετρο -u. Τέλος ανήκει στην ομάδα χρηστών του ΥΔΡΟΣΚΟΠΙΟΥ hdba και έχει τα αντίστοιχα προνόμια.

- **Διαχειριστής συστήματος root.** Πρόκειται βέβαια για τον κωδικό του διαχειριστή του συστήματος (σταθμού εργασίας). Έχει πλήρη δικαιώματα ανάγνωσης και εγγραφής σε όλα τα αρχεία του συστήματος. Μπορεί επίσης να χρησιμοποιήσει το ΣΔΒΔ και να προσπελάσει τις ΒΔ του ΥΔΡΟΣΚΟΠΙΟΥ db0 και ddb0/d. Εξ' ορισμού ο χρήστης root ανήκει στην ομάδα hdba.
- **Εξ' ορισμού κωδικοί χρήστη σε απομακρυσμένα συστήματα** ntua0, nms0, ma0, meppw0, noa0, prc0, wssca0, ua0, miet0, autdhee0, autde0, nrpsd0. Οι κωδικοί αυτοί αντιστοιχούν στους Φορείς ΕΜΠ, ΕΜΥ, ΥΠΓΕ, ΥΠΕΧΩΔΕ, ΕΑΑ, ΔΕΗ, ΕΥΔΑΠ, ΕΚΠΑ, ΥΒΕΤ, ΑΠΘ/ΤΥΤΠ, ΑΠΘ/ΕΤ και ΕΚΕΦΕ"Δ" και είναι αυτοί στους οποίους αντιστοιχίζονται όλοι οι κωδικοί του Φορέα προκειμένου για πρόσβαση στον έξω κόσμο. Οι κωδικοί αυτοί ανήκουν σε όλα τα συστήματα στην ομάδα hrusers. Για παράδειγμα, ο χρήστης hydro του κόμβου του ΥΠΕΧΩΔΕ έχει πλήρη εξουσία στην δική του ΤΒΔ. Κάθε όμως πρόσβαση σε ΤΒΔ άλλων κόμβων δεν γίνεται ως hydro αλλά ως meppw0 που ανήκει σε όλους τους κόμβους στην ομάδα περιορισμένων δικαιωμάτων hrusers. Αυτό γίνεται δυνατό μέσω του μηχανισμού καθορισμού κωδικών του INGRES/NET και του προγράμματος netu [INGRES91c].
- **Εξ' ορισμού κωδικοί χρήστη αντιγραφής σε απομακρυσμένα συστήματα** ntua8, nms8, ma8, meppw8, noa8, prc8, wssca8, ua8, miet8, autdhee8, autde8, nrpsd8. Οι κωδικοί αυτοί είναι αντίστοιχοι με τους προηγούμενους αλλά αναφέρονται στους δαίμονες αντιγραφής hrpld και ανήκουν σε όλα τα συστήματα στην ομάδα hrpl ώστε να έχουν δικαιώματα εγγραφής στους admin_rmt πίνακες. Αυτό σημαίνει πως, ενώ όλοι οι "κανονικοί" χρήστες ενός κόμβου επικοινωνούν με τις ΒΔ των άλλων κόμβων με χρήση του εξ' ορισμού κωδικού χρήστη του Φορέα του κόμβου, πχ. meppw0 για το ΥΠΕΧΩΔΕ, με τα περιορισμένα δικαιώματα της ομάδας hrusers, οι δαίμονες αντιγραφής επικοινωνούν με τις ΒΔ των άλλων κόμβων με χρήση αυτών των κωδικών που ανήκουν στην ειδική ομάδα hrpl. Η ομάδα αυτή έχει τα ίδια δικαιώματα με το κοινό public εκτός των πινάκων admin_rmt όπου έχει πλήρη δικαιώματα, ώστε να μπορεί κάθε δαίμονας αντιγραφής να τροποποιήσει τα δεδομένα των πινάκων αυτών, που είναι και το ζητούμενο.

4.4.1 Δημιουργία κωδικών χρήστη. Άλλοι κωδικοί εκτός από τους εξ' ορισμού μπορούν και πρέπει να δημιουργηθούν σε κάθε κόμβο. Κάθε χρήστης που επιθυμεί προσπέλαση στο ΥΔΡΟΣΚΟΠΙΟ αποκτά έναν κωδικό χρήστη και στη συνέχεια προστίθεται στον κατάλογο των νόμιμων χρηστών του ΣΔΒΔ INGRES και του ΥΔΡΟΣΚΟΠΙΟΥ. Προστίθεται σε όσες από τις ομάδες θεωρείται πως πρέπει να συμμετέχει, ανάλογα με τα καθήκοντα του στο Φορέα. Για παράδειγμα ένα πρόσωπο

επιφορτισμένο γενικά με τον έλεγχο των εισαγόμενων στοιχείων μπορεί να ανήκει στις ομάδες `hdataupdate` και `hadminupdate`. Κάθε σύνδεση με τη ΒΔ (πχ. εκτέλεση της εφαρμογής του ΥΔΡΟΣΚΟΠΙΟΥ) γίνεται με μια ομάδα. Για να αλλάξει ομάδα ο χρήστης πρέπει εκ νέου να συνδεθεί με τη ΒΔ. Για να αλλάξει δε κωδικό χρήστη πρέπει είτε να χρησιμοποιήσει την εντολή `su` του UNIX είτε να ζαναπροσπελάσει το σύστημα με το νέο κωδικό. Οποσδήποτε πρέπει να αποφευχθεί η συνεχής χρήση της ΒΔ από τον κωδικό `hydro`. Τέλος, ο διαχειριστής ΒΔ ενός κόμβου έχει το δικαίωμα να δώσει και σε έναν απομακρυσμένο χρήστη περισσότερα δικαιώματα πρόσβασης απ' όσα του δίνει η εξ' ορισμού συμμετοχή του στην ομάδα `husers`. Αρκεί να δημιουργήσει κάποιο κωδικό για τον απομακρυσμένο χρήστη στο τοπικό σύστημα οπότε όλο που έχει να κάνει ο χρήστης αυτός στο δικό του σύστημα είναι να χρησιμοποιήσει το `netu` για να εισάγει μια ιδιωτική εγγραφή { Γ }.

Νέοι χρήστες εισάγονται στο σύστημα από τον κωδικό `root` με χρήση της διαλογικής εντολής `addunixuser`. Νέοι χρήστες εισάγονται στην INGRES και το ΥΔΡΟΣΚΟΠΙΟ από τον χρήστη `ingres` με χρήση της διαλογικής εντολής `addingreshydrouser`. Η χρήση των εντολών αυτών είναι εξαιρετικά απλή, μια και όλο που πρέπει να γίνει είναι να απαντηθούν οι ερωτήσεις που θέτει το πρόγραμμα.

4.5 Πίνακες ΒΔ για το υποσύστημα ασφάλειας

Οι πίνακες της ΒΔ `db0` που υποστηρίζουν το σύστημα ασφάλειας είναι οι ακόλουθοι:

- **userid**s. Ο πίνακας αυτός περιέχει, στο `char(32)` πεδίο `userid`, τους υπάρχοντες χρήστες του ΥΔΡΟΣΚΟΠΙΟΥ στον τοπικό κόμβο.
- **groupid**s. Ο πίνακας αυτός περιέχει, στο `char(32)` πεδίο `groupid`, τις υπάρχουσες ομάδες. Προφανώς σε όλους τους κόμβους υπάρχουν οι ομάδες `husers`, `husers`, `hdatainsert`, `hdataupdate`, `hadmininsert`, `hadminupdate`, `hrpl`, `hdba` και `hdev`.
- **groupid_access_ts**. Ο πίνακας αυτός περιέχει, στο `char(32)` πεδίο `groupid_access_t` τις δυνατές ενέργειες πάνω στα αντικείμενα της ΒΔ ανάλογα με τον τύπο τους. Συγκεκριμένα, οι ενέργειες αυτές είναι `select` (για ανάκτηση πληροφοριών), `datainsert` (για εισαγωγή πληροφοριών σε πίνακες `data`), `dataupdate` (για τροποποίηση -εισαγωγή, μεταβολή, διαγραφή- πληροφοριών σε πίνακες `data`), `admininsert` (για εισαγωγή πληροφοριών σε πίνακες `admin`), `adminupdate` (για τροποποίηση πληροφοριών σε πίνακες `admin`), `rmtupdate` (για τροποποίηση πληροφοριών από απομακρυσμένους κόμβους σε πίνακες `admin_rmt` -για χρήση από τους δαίμονες αντιγραφής).

- **groupid_access.** Σε σχέση με τις δυνατές ενέργειες του πίνακα `groupid_access_ts` ο πίνακας `groupid_access` περιγράφει ποιές από αυτές τις ενέργειες μπορούν να εκτελέσουν οι διάφορες ομάδες που περιέχονται στον πίνακα `groupids`. Πρακτικά δηλαδή καταγράφει σε πίνακες αυτό που το ΣΔΒΔ γνωρίζει μέσω των εντολών `grant`: ποιά δικαιώματα πρόσβασης και σε ποιά αντικείμενα (ακριβέστερα, σε ποιές κατηγορίες αντικειμένων) έχει κάθε ομάδα. Ο πίνακας έχει δύο πεδία, το `char(32) groupid` που περιέχει το όνομα της ομάδας (`groupids` είναι βέβαια ο πίνακας αναφοράς) και το `char(32) groupid_access_t` (με πίνακα αναφοράς τον `groupid_access_ts`) που περιέχει το είδος πρόσβασης - δυνατές ενέργειες της ομάδας. Έτσι ένα ζεύγος πχ. `hdataupdate, dataupdate` σημαίνει πως η ομάδα `hdataupdate` έχει δικαίωμα `update` (ακριβέστερα: `update & delete & insert`) στους πίνακες `data`.
- **userid_groupids.** Ο πίνακας αυτός περιέχει εγγραφές με ζευγάρια χρηστών - ομάδων. Οι χρήστες περιέχονται στο `char(32)` πεδίο `userid` (προφανώς με πίνακα αναφοράς τον `userid`) και οι ομάδες στο `char(32)` πεδίο `groupid` (με πίνακα αναφοράς τον `groupids`). Η ύπαρξη ενός ζεύγους χρήστη - ομάδας σημαίνει πως ο χρήστης ανήκει στην ομάδα. Πχ. η εγγραφή `user1, hdatainsert` σημαίνει πως ο χρήστης `user1` ανήκει στην ομάδα `hdatainsert`.

Εισαγωγή δεδομένων στους πίνακες `userid` και `userid_groupids` μπορούν να γίνουν είτε "χειρωνακτικά" είτε αυτόματα κατά την εισαγωγή του νέου κωδικού χρήστη από το `addingreshydrouser`.

4.6 Ασφάλεια εφαρμογών

Το σύστημα ασφάλειας λειτουργεί αυτόματα στη ΒΔ. Δηλαδή, το ΣΔΒΔ φροντίζει μόνο του να ρυθμίζει την προσπέλαση στα διάφορα αντικείμενα της ΒΔ ώστε οι κατάλληλοι χρήστες να έχουν τα κατάλληλα δικαιώματα πρόσβασης. Χρειάζεται ωστόσο και αντίστοιχη πρόνοια στο επίπεδο της εφαρμογής. Μια εφαρμογή που έχει τη δυνατότητα τροποποίησης κάποιων αντικειμένων της ΒΔ δεν πρέπει να δίνει στο χρήστη την εντύπωση πως έχει δικαίωμα τροποποίησης αν η ομάδα στην οποία ανήκει ο χρήστης δεν έχει αυτό το δικαίωμα. Αν αυτό ήταν δυνατό ο χρήστης που θα προσπαθούσε να τροποποιήσει κάποια αντικείμενα χωρίς το αντίστοιχο δικαίωμα πρόσβασης θα διαπίστωνε εκ των υστέρων με κάποιο κατάλληλο μήνυμα πως το ΣΔΒΔ του απαγορεύει την πρόσβαση. Αυτή η πολιτική ασφαλώς δεν είναι φιλική για το χρήστη και για να αποφευχθεί πρέπει το κουμπί { Γ } ή η επιλογή του μενού { Γ } που ξεκινά τη διαδικασία τροποποίησης να είναι σε περίπτωση απουσίας δικαιώματος είτε

“αχνό”{ Γ } είτε αόρατο { Γ } ώστε ο χρήστης να μην μπορεί να το επιλέξει. Για να γίνει αυτό ακολουθείται η εξής μεθοδολογία¹⁰:

Στην εφαρμογή του ΥΔΡΟΣΚΟΠΙΟΥ υπάρχει μια παγκόσμια μεταβλητή, η GlobalData η οποία περιέχει όλα τα απαραίτητα στοιχεία που πρέπει να είναι γνωστά σε όλα τα τμήματα της εφαρμογής για τη σωστή λειτουργία της. Κάθε αντικείμενο σε μια φόρμα { Γ } των Windows/4GL [INGRES91e][INGRES91f] έχει μια ιδιότητα { Γ }, την CurBias που καθορίζει με ποιά μορφή το αντικείμενο θα φαίνεται στην οθόνη και ποιά θα είναι η συμπεριφορά του σε σχέση με τις ενέργειες του χρήστη. Αν πχ. το αντικείμενο είναι κουμπί η τιμή της ιδιότητας FB_LANDABLE σημαίνει πως ο χρήστης μπορεί να το “πιέσει” και να ενεργοποιήσει έτσι τη διαδικασία που αντιστοιχεί σε αυτό, η τιμή FB_DIMMED σημαίνει πως το αντικείμενο - κουμπί είναι “αχνό” δηλαδή φαίνεται στην οθόνη αλλά δεν μπορεί να πιεσθεί, ενώ η τιμή FB_INVISIBLE σημαίνει πως το αντικείμενο είναι αόρατο στο χρήστη. Αντίστοιχα, οι επιλογές των μενού έχουν χαρακτηριστικές τιμές της CurBias ιδιότητας τους τις MB_ENABLED που σημαίνει πως η επιλογή είναι διαθέσιμη στο χρήστη, MB_DISABLED που σημαίνει πως είναι αχνό και MB_INVISIBLE που σημαίνει πως είναι αόρατη. Προφανώς η ιδιότητα CurBias είναι πολύ βολική για την ενεργοποίηση του συστήματος ασφαλείας στις εφαρμογές: αν η ενέργεια που σηματοδοτείται από το αντικείμενο της φόρμας - κουμπί ή την επιλογή του μενού επιτρέπεται από το σύστημα ασφαλείας της ΒΔ τότε το αντικείμενο ή η επιλογή μπορούν να έχουν CurBias = FB_LANDABLE ή MB_ENABLED. Διαφορετικά θα είναι είτε αόρατα είτε αχνό. Για παράδειγμα, αν ένα κουμπί ενεργοποιεί “σώσιμο” νέων δεδομένων στη ΒΔ θα πρέπει για να μπορεί να ενεργοποιηθεί, δηλαδή για να έχει CurBias = FB_LANDABLE ο χρήστης να ανήκει στην ομάδα hdatainsert. Αν δεν ανήκει το κουμπί πρέπει να είναι αχνό, FB_DIMMED. Μένει λοιπόν να γνωρίζει η εφαρμογή σε ποιά ομάδα έχει γίνει η σύνδεση του χρήστη στην εφαρμογή και ποιά είναι τα δικαιώματα αυτής της ομάδας πάνω σε κάθε τύπο δεδομένων. Το πρώτο είναι απλό: ο χρήστης μπορεί στο αρχικό πλαίσιο { Γ } “Σύνδεση” της εφαρμογής ΥΔΡΟΣΚΟΠΙΟΥ να δώσει κωδικό ομάδας στην οποία επιθυμεί να ανήκει για το διάστημα της σύνδεσης του. Αν δε δώσει κωδικό θα ανήκει στην εξ’ ορισμού ομάδα του (συνήθως husers). Σε κάθε περίπτωση η ομάδα περιέχεται στο πεδίο GlobalData.GroupName. Δύο άλλα πεδία της GlobalData, τα FieldAccess και MenuAccess είναι τύπου GlobalDataAccessClass και περιέχουν το καθένα 4 integer4 υποπεδία, τα DataInsert, DataUpdate, AdminInsert και AdminUpdate. Τα υποπεδία ταυτίζονται με τις κυριότερες δυνατές ενέργειες πάνω στα αντικείμενα της ΒΔ όπως φαίνονται και στον πίνακα groupid_access_ts. Η τιμή τους

10 Στη συνέχεια του κειμένου σε όσα σημεία γίνεται αναφορά σε εφαρμογή αυτή υποτίθεται πως είναι γραμμένη στη βασική γλώσσα ανάπτυξης εφαρμογών του ΥΔΡΟΣΚΟΠΙΟΥ, την INGRES Windows/4GL. Τυχόν διαφοροποιήσεις θα επισημαίνονται κατάλληλα.

είναι η "ορδή" τιμή του `CurBias` για αντικείμενα της φόρμας και του μενού αντίστοιχα. Με το "ορδή" εννοείται ποιά τιμή πρέπει να πάρει το `CurBias` ώστε από τα αντίστοιχα αντικείμενα της φόρμας ή του μενού να είναι ενεργές ή ανενεργές οι ενέργειες στη ΒΔ ανάλογα με την ομάδα του χρήστη. Δηλαδή, αν κάποιος χρήστης ανήκει σε μια ομάδα που έχει δικαίωμα `DataInsert` τότε θα γίνει `GlobalData.FieldAccess.DataInsert = FB_LANDABLE` και `GlobalData.MenuAccess.DataInsert = MB_ENABLED`. Αν δεν έχει τέτοιο δικαίωμα τότε θα γίνει `GlobalData.FieldAccess.DataInsert = FB_DIMMED` και `GlobalData.MenuAccess.DataInsert = MB_DISABLED`. Η αρχικοποίηση { Γ } των υποπεδίων γίνεται στη συνάρτηση αρχικοποίησης του `GlobalData`, `GlobalDataInit()` με κώδικα σαν τον παρακάτω:

```

...
/* Ελεγχος αν η ομάδα του χρήστη έχει δικαίωμα datainsert
*/
select   :r = count(*)
from groupid_access
where    groupid_access_t = 'datainsert'
        and
        groupid = :GlobalData.GroupName;
if r is not null and r > 0 then
    /* Η ομάδα έχει όντως δικαίωμα datainsert, άρα τα
    αντικείμενα που κάνουν αυτή την ενέργεια
    πρέπει να είναι ενεργοποιημένα */
    GlobalData.FieldAccess.DataInsert = FB_LANDABLE;
    GlobalData.MenuAccess.DataInsert = MB_ENABLED;
else
    /* Η ομάδα δεν έχει δικαίωμα datainsert, άρα τα
    αντικείμενα που κάνουν αυτή την ενέργεια
    πρέπει να είναι απενεργοποιημένα */
    GlobalData.FieldAccess.DataInsert = FB_DIMMED;
    GlobalData.MenuAccess.DataInsert = MB_DISABLED;
endif;

```

Για παράδειγμα, αν κάποιο αντικείμενο - κουμπί της φόρμας, έστω `MyButton`, οδηγεί με την επιλογή - πάτημα του σε ενέργεια εισαγωγής δεδομένων `datainsert` τότε ο προγραμματιστής αρκεί να δέσει στο τμήμα `initialize` της φόρμας την εντολή `field(MyButton).CurBias = GlobalData.FieldAccess.DataInsert;` και αυτόματα (λόγω της κατάλληλης αρχικοποίησης στην `GlobalDataInit()`) το αντικείμενο παίρνει την ορδή τιμή του `CurBias` σε σχέση με την ομάδα στην οποία ανήκει ο χρήστης: `FB_LANDABLE` (και το κουμπί ενεργοποιείται) αν η ομάδα του χρήστη έχει δικαίωμα `datainsert` και `FB_DIMMED` (και το κουμπί απενεργοποιείται και γίνεται αχνό) αν δεν έχει. Για να εισαχθούν λοιπόν χαρακτηριστικά ασφάλειας σε όλες τις εφαρμογές αρκεί να καθορισθεί ποιά από τις 4 βασικές ενέργειες προκαλεί κάθε αντικείμενο της φόρμας - κουμπί και επιλογή μενού και να δοθούν εντολές σαν την παραπάνω (για τα μενού πρέπει βέβαια να χρησιμοποιηθεί το πεδίο `GlobalData.MenuAccess`).

4.7 Γενικά για το υποσύστημα χρέωσης

Αντίθετα με το υποσύστημα ασφάλειας που υλοποιείται κατά κύριο λόγο μέσα στο ΣΔΒΔ και τη ΒΔ και επικουρικά μόνο στις εφαρμογές, το υποσύστημα χρέωσης πρέπει να υλοποιηθεί στο επίπεδο των εφαρμογών μια και το ΣΔΒΔ δεν παρέχει ευκολίες για τη λειτουργία της. Θα έπρεπε για παράδειγμα να είναι δυνατόν να ορίζονται κανόνες ΒΔ { Γ } και για την πράξη `select` εκτός από τις πράξεις `insert`, `update`, `delete`. Σε αυτή την περίπτωση η χρέωση θα περιοριζόταν στην ενεργοποίηση ενός κανόνα για κάθε ανάκτηση δεδομένων. Τέτοια δυνατότητα ωστόσο δεν υπάρχει και γι' αυτό το λόγο οι εφαρμογές θα πρέπει να φροντίσουν για την υλοποίηση της χρέωσης. Τα κύρια χαρακτηριστικά του συστήματος χρέωσης είναι τα ακόλουθα [Πιπ92β]:

- Χρέωση υπάρχει μόνο για ανάκτηση (`select`) δεδομένων και ειδικότερα μόνο για ανάκτηση χρονοσειρών και των σταθερών. Δεν υπάρχει χρέωση για τα διαχειριστικά δεδομένα και τα δεδομένα των εφαρμογών.
- Η χρέωση είναι διαφορετική ανάλογα με το αν τα δεδομένα μεταφέρονται πάνω από το δίκτυο ή είναι τοπικά. Τα απομακρυσμένα δεδομένα είναι ακριβότερα.
- Η χρέωση είναι ευθέως ανάλογη και γραμμικά εξαρτημένη με δύο κύριες παραμέτρους: η μία είναι ο αριθμός των εγγραφών που επεστράφησαν από την ενέργεια ανάκτησης και η άλλη ο ολικός χρόνος επεξεργασίας της ερώτησης από τη στιγμή της υποβολής της ως τη στιγμή της ολοκλήρωσης της.
- Βοηθητικές παράμετροι που επηρεάζουν την χρέωση είναι: το επίπεδο (1, 2, 3) των δεδομένων που ικανοποιούν την ερώτηση, το είδος τους (τα ΠΡΩΤΟΓ είναι "φτηνότερα" από τα ΠΑΡΑΓ και τα ΔΕΥΤΕΡΟΓ), η χρονική κλίμακα τους (τα ΩΡΙΑΙΑ δεδομένα είναι "φτηνότερα" από τα ΕΤΗΣΙΑ μια και έχουν υποστεί πολύ μικρότερη επεξεργασία. Από την άλλη βέβαια τα ΩΡΙΑΙΑ δεδομένα είναι πολύ περισσότερα σε πλήθος, αντισταθμίζοντας το παραπάνω πλεονέκτημα), το είδος του μετρητικού οργάνου (τα αυτογραφικά όργανα έχουν "ακριβότερες" χρονοσειρές από τα απλά και τα στοιχεία ποιότητας "ακριβότερες" και από τα δύο), η μορφή της απάντησης (τα δεδομένα στην οδόνη είναι φτηνότερα από αυτά στον εκτυπωτή) και τέλος το αν πρόκειται για ειδική πληροφορία (πχ. σταθερές σταθμών ΥΥΥ).
- Κάθε σύνδεση με τη ΒΔ είναι το βασικό περιβάλλον μέσα στο οποίο λειτουργεί το σύστημα χρέωσης. Χαρακτηριστικά της σύνδεσης είναι ο κωδικός χρήστη και η ομάδα της, η ώρα που ξεκίνησε κοκ. Επίσης σε κάθε σύνδεση δίνεται ένας αύξων αριθμός.
- Στο πλαίσιο μιας σύνδεσης κάθε ενέργεια ανάκτησης χρονοσειρών ή σταθερών χρεώνεται ξεχωριστά.

- Η παρούσα υλοποίηση του συστήματος χρέωσης αποθηκεύει όλες τις παραμέτρους της χρέωσης σε κάποιους πίνακες. Καθώς δεν έχει γίνει ακόμα συγκεκριμένος ο αλγόριθμος της χρέωσης του ΥΔΡΟΣΚΟΠΙΟΥ δεν παρέχονται “τελικές τιμές”. Ο υπολογισμός τους είναι εύκολος με βάση τις φυλασσόμενες παράμετρους. Αρκεί ένα απλό πρόγραμμα να διαβάσει τους πίνακες και να εκδίδει στοιχεία χρέωσης.

4.8 Πίνακες και διαδικασίες ΒΔ για το υποσύστημα χρέωσης

Βασικό χαρακτηριστικό του υποσυστήματος χρέωσης από την πλευρά της ΒΔ είναι πως η ενημέρωση των πινάκων του γίνεται από μια διαδικασία ΒΔ, την `_acct()`. Ο λόγος για αυτήν την πρακτική είναι απλός: όλοι οι κωδικοί χρηστών θα πρέπει να ενημερώνουν τα στοιχεία χρέωσης που φυλάσσονται σε κάποιους πίνακες. Προφανώς όμως τυχόν δικαίωμα απευθείας πρόσβασης σε αυτούς θα έθετε σε κίνδυνο την ορθή λειτουργία του συστήματος (ορισμένοι χρήστες θα ανακάλυπταν πως μπορούν να σθήσουν τα στοιχεία χρέωσης που τους βαρύνουν). Για το λόγο αυτό, οι πίνακες δεν έχουν κανένα δικαίωμα πρόσβασης, ούτε καν `select`, για κανέναν, εκτός από την ομάδα `hdba` και βέβαια το διαχειριστή ΒΔ `hydro`. Έτσι δεν μπορεί ένας χρήστης να διαπιστώσει τη χρήση του συστήματος από κάποιον άλλον. Σε όλους τους χρήστες όμως δίνεται άδεια εκτέλεσης της διαδικασίας ΒΔ `_acct()` η οποία με τη σειρά της ενημερώνει τους πίνακες.

Η `_acct()` καλείται στις εξής περιπτώσεις:

- όταν γίνεται μια νέα σύνδεση στη ΒΔ για να την αρχικοποιήσει, να καταγράψει την ώρα έναρξης, τους κωδικούς χρήστη και ομάδας κτλ. και να επιστρέψει τον χαρακτηριστικό αύξοντα αριθμό της σύνδεσης. Στην περίπτωση αυτή η `_acct()` καλείται ως εξής:

```

SessionId = callproc _acct(
    _what = 'open',
    userid = :user, /* Ο κωδικός χρήστη στο
                    τοπικό σύστημα */
    groupid = :group, /* Η ομάδα */
    rmtuserid = :rmtuser, /* Τυχόν
                          διαφορετικός κωδικός
                          χρήστη για
                          απομακρυσμένη πρόσβαση
                          */
    rmtgroupid = :rmtgroup /* Τυχόν
                          διαφορετική ομάδα για
                          απομακρυσμένη πρόσβαση
                          (δεν χρησιμοποιείται)
                          */
);

```

- όταν τελειώνει μια σύνδεση με τη ΒΔ για να καταγράψει την ώρα τερματισμού της. Η κλήση έχει τότε ως εξής:

```

SessionId = callproc _acct(
    _what = 'close',
    userid = :user,
    groupid = :group,
    rmtuserid = :rmtuser,
    rmtgroupid = :rmtgroup,
    session = :SessionId /* Ο α/α σύνδεσης
                           που είχε επιστραφεί
                           κατά την έναρξη της
                           σύνδεσης */
);

```

- αμέσως πριν αρχίσει μια ενέργεια ανάκτησης δεδομένων για να καταγράψει την ώρα έναρξης της ώστε να είναι δυνατό να βρεθεί ο χρόνος που απαιτήθηκε για την εκτέλεση της ερώτησης. Η κλήση γίνεται ως ακολούθως:

```

StartTime = callproc _acct(
    _what = 'start',
    userid = :user,
    groupid = :group,
    rmtuserid = :rmtuser,
    rmtgroupid = :rmtgroup
);

```

Η τιμή που επιστρέφεται είναι ο αριθμός των δευτερολέπτων που είχαν περάσει στη σύνδεση μέχρι τη στιγμή της κλήσης. Αμέσως μετά πρέπει να ακολουθεί η ανάκτηση των δεδομένων.

- αμέσως μετά το πέρας μιας ενέργειας ανάκτησης καλείται η `_acct()` για να καταγράψει τα στοιχεία της ανάκτησης που μόλις τελείωσε. Υποτίθεται πως οι εγγραφές έχουν τοποθετηθεί σε κάποιου είδους `ArrayObject` ή `TableField`, έστω `A`, άδαιο πριν την ανάκτηση οπότε η μέθοδος `A.AllRows()` δίνει τον αριθμό των εγγραφών που ανακτήθηκαν. Επίσης η εφαρμογή είναι σε δέση να γνωρίζει τις διάφορες "βοηθητικές" παραμέτρους της χρέωσης (είδος δεδομένων, χρονική κλίμακα κοκ). Έτσι γίνεται:

```

callproc _acct(
    _what = 'end', /* Αν το _what έχει τιμή 'fail'
                   τότε σημαίνει πως η ανάκτηση για
                   κάποιο λόγο απέτυχε οπότε δεν
                   γίνεται χρέωση. Το ίδιο και αν η
                   παράμετρος η παρακάτω είναι null ή
                   < 0 */
    userid = :user,
    groupid = :group,
    rmtuserid = :rmtuser,
    rmtgroupid = :rmtgroup,
    session = :SessionId, /* Ο α/α της σύνδεσης */
    n = :A.AllRows(), /* Ο αριθμός των εγγραφών που
                      ανακτήθηκαν */
    t = :StartTime, /* Πότε άρχισε η ανάκτηση ώστε
                    να γίνει η αφαίρεση από την
                    τρέχουσα χρονική στιγμή και να
                    διαπιστωθεί ο "καθαρός" χρόνος

```

```

        της ερώτησης */
level = :level, /* 1, 2, 3 ή 0 */
distr = :distr, /* 1 αν η πρόσβαση ήταν
        κατανεμημένη (πάνω από το δίκτυο, 0
        διαφορετικά */
scale = :scale, /* Η χρονική κλίμακα των δεδομένων.
        Παίρνει τιμές 'overyear' ή
        'ΥΠΕΡΕΤΗΣ', 'year' ή 'ΕΤΗΣΙΑ',
        'month' ή 'ΜΗΝΙΑΙΑ', 'day' ή
        'ΗΜΕΡΗΣΙΑ', 'hour' ή 'ΩΡΙΑΙΑ',
        'subhour' ή 'ΥΠΩΡΙΑΙΑ', null ή '' ή
        'ΑΓΝΩΣΤΟ' */
instr = :instr, /* Το είδος του οργάνου. Παίρνει
        τιμές 'normal' ή 'ΚΑΝΟΝΙΚΟ',
        'auto' ή 'ΑΥΤΟΓΡΑΦ', 'quality' ή
        'ΠΟΙΟΤΗΤΑ', null ή '' ή
        'ΑΓΝΩΣΤΟ' */
type = :type, /* Το είδος των δεδομένων. Παίρνει
        τιμές 'raw' ή 'ΠΡΩΤΟΓ', 'derived' ή
        'ΠΑΡΑΓ', 'aggr' ή
        'ΔΕΥΤΕΡΟΓ', 'minmax' ή 'ΜΕΓ-ΕΛΑΧ'
        (μέγιστα - ελάχιστα), 'extra' ή
        'ΔΙΑΦΟΡΑ', null ή '' ή 'ΑΓΝΩΣΤΟ' */
display = :display, /* Η μορφή της απάντησης.
        Παίρνει τιμές 'display' ή
        'ΘΟΟΝΗ', 'ascii' ή
        'ΜΑΓΝΗΤ', 'print' ή 'ΕΚΤΥΠΩΣΗ',
        'bitmap' ή 'ΕΙΔΙΚΗ', null ή ''
        ή 'ΑΓΝΩΣΤΟ' */
special = :special /* Ειδική πληροφορία. Παίρνει
        τιμές 'station' ή 'ΣΤΑΘΜΟΣ',
        'drilling' ή 'ΟΠΗ', 'piping' ή
        'ΣΩΛΗΝΩΣΗ', 'pumptest' ή
        'ΔΟΚ-ΑΝΤΑ', 'lithsect' ή
        'image' ή 'ΕΙΚΟΝΑ', null ή '' ή
        'ΑΓΝΩΣΤΟ' */
);

```

Οι πίνακες που σχετίζονται με τη χρέωση είναι οι ακόλουθοι:

- **acct_ts.** Περιέχονται στο char (8) πεδίο acct_t όλες οι ειδικές παράμετροι όπως φάνηκαν στην περιγραφή της acct(). Πχ. 'ΕΙΚΟΝΑ', 'ΠΑΡΑΓ' κοκ.
- **acct_sessions.** Περιέχονται τα στοιχεία κάθε σύνδεσης με τη ΒΔ μέσα από την εφαρμογή του ΥΔΡΟΣΚΟΠΙΟΥ. Τα στοιχεία αυτά είναι ο char(32) τοπικός κωδικός χρήστη userid (πίνακας αναφοράς userids), η char(32) ομάδα groupid (πίνακας αναφοράς groupids), ο char(32) απομακρυσμένος κωδικός χρήστη rmtuserid και η αντίστοιχη ομάδα rmtgroupid, η ημερομηνία - ώρα αρχής και τέλους αντίστοιχα της σύνδεσης, start_session και end_session τύπου date και ο α/α της σύνδεσης για αυτό τον τοπικό κωδικό χρήστη, session που είναι τύπου integer4. Υποτίθεται πως το απλό πρόγραμμα έκδοσης στοιχείων

χρέωσης θα σβήνει σε κάποια χρονικά διαστήματα τα περιεχόμενα του πίνακα αυτού όπως και του `acct_session_transcript` για να μην καταλάβουν υπερβολικά πολύ χώρο.

- `acct_session_transcript`. Για κάθε σύνδεση με τη ΒΔ περιέχονται λεπτομερείς πληροφορίες για τις ανακτήσεις δεδομένων που έγιναν στη διάρκεια της. Τα πεδία του είναι το `char(32) userid` που είναι ο τοπικός κωδικός χρήστη στον οποίο αντιστοιχεί η `integer4` σύνδεση `session`. Η ημερομηνία `start_time` τύπου `date` είναι η στιγμή έναρξης της ανάκτησης δεδομένων. Τα `integer4` πεδία `t` και `n`, τα `integer1` πεδία `level` και `distr` και τα `char(8)` πεδία `scale`, `instr`, `type`, `display` και `special` έχουν ακριβώς τον ίδιο τύπο, νόημα και δυνατές τιμές με αυτές που παρουσιάζονται στην περιγραφή της `_acct()`. Προφανώς τότε ο πίνακας αυτός έχει πλήρεις πληροφορίες για την συγκεκριμένη ανάκτηση δεδομένων και σε συνδυασμό με τα στοιχεία του `acct_sessions` μπορεί να δώσει πλήρεις πληροφορίες για την εξέλιξη κάθε σύνδεσης στο σύστημα.

4.9 Χρέωση εφαρμογών

Οι εφαρμογές δεν χρειάζεται να καλούν απευθείας την `_acct()`. Αρκεί πριν την ανάκτηση δεδομένων να καλούν τη ρουτίνα `BeforeQuery()` και μετά την `AfterQuery()`, δηλαδή

```
callproc BeforeQuery(...);
select ...
callproc AfterQuery(...);
```

Η χρήση των ρουτινών αυτών περιγράφεται στο επόμενο κεφάλαιο.

5 ΥΠΟΣΤΗΡΙΞΗ ΕΦΑΡΜΟΓΩΝ

5.1 Συναρτήσεις BeforeQuery() και AfterQuery()

Η χρήση των συναρτήσεων BeforeQuery() και AfterQuery() έχει δύο κύριους σκοπούς:

- **Αλλαγή ενεργής σύνδεσης ΚΒΔ / ΤΒΔ.** Κάθε εφαρμογή διατηρεί δύο παράλληλες συνδέσεις. Η GlobalData.LocalDBSession είναι μια σύνδεση με την ΤΒΔ και η GlobalData.DistrDBSession μια σύνδεση με την ΚΒΔ. Κατά τη μεγαλύτερη διάρκεια της λειτουργίας τους οι εφαρμογές χρησιμοποιούν τη σύνδεση με την ΤΒΔ για να προσπελάσουν διαχειριστικές πληροφορίες που θα δώσουν στο χρήστη μια εικόνα της διαθεσιμότητας των δεδομένων που ζητά, καθώς και πληροφορίες εφαρμογών. Μόλις πριν την έναρξη της ανάκτησης των δεδομένων χρειάζεται η εφαρμογή να αλλάξει την ενεργή σύνδεση και να χρησιμοποιήσει τη σύνδεση με την ΚΒΔ ώστε να φέρει δεδομένα από το δίκτυο. Εννοείται φυσικά πως υπάρχει και η περίπτωση ο χρήστης να επιλέξει, από κατάλληλη επιλογή μενού του κύριου μενού της εφαρμογής να προσπελάσει μόνο την ΤΒΔ για επεξεργασία τοπικών στοιχείων. Σε κάθε περίπτωση η ενεργή σύνδεση παραμένει αυτή της ΚΒΔ όσο διαρκεί η ανάκτηση των πληροφοριών. Στη συνέχεια πρέπει η ενεργή σύνδεση να επανέλθει στην ΤΒΔ ώστε να συνεχιστεί η κανονική λειτουργία της εφαρμογής.
- **Χρέωση.** Οι δύο συναρτήσεις εκτελούν για λογαριασμό του προγραμματιστή εκτός από την αλλαγή της ενεργής σύνδεσης και τις διαδικασίες της χρέωσης όπως περιγράφησαν στο προηγούμενο κεφάλαιο. Ο προγραμματιστής δεν χρειάζεται να ασχοληθεί με τις λεπτομέρειες (κλήση _acct() με κατάλληλες παραμέτρους κτλ.), αρκεί να περάσει ως παραμέτρους τα στοιχεία της ανάκτησης δεδομένων.

Ειδικότερα, η BeforeQuery() πρέπει να χρησιμοποιείται ακριβώς πριν την έναρξη της ανάκτησης δεδομένων και η AfterQuery() ακριβώς μετά. Στις άλλες ενέργειες SQL δηλαδή insert, update, delete δεν χρειάζεται η κλήση των συναρτήσεων. Καταρχήν διότι αυτές οι ενέργειες γίνονται πάντα στην ΤΒΔ και ποτέ πάνω από το δίκτυο (δεν είναι δυνατή η τροποποίηση πληροφοριών σε άλλους κόμβους - εκτός από την ειδική περίπτωση της αντιγραφής διαχειριστικών πληροφοριών που όμως γίνεται εκτός όλων των εφαρμογών από τον ειδικό δαίμονα αντιγραφής) και επίσης διότι χρέωση υφίσταται μόνο στην ανάκτηση δεδομένων.

5.1.1 Συνάρτηση BeforeQuery(). Η συνάρτηση αυτή έχει σαν αποστολή την αλλαγή της ενεργής σύνδεσης από την ΤΒΔ στην ΚΒΔ και επίσης την έναρξη της διαδικασίας χρέωσης με κατάλληλη κλήση _acct(_what = 'start', ...). Υπάρχει όμως και ένα σοβαρό σημείο που χρήζει προσοχής. Οι περισσότερες ερωτήσεις των εφαρμογών του


```

                                χρησιμοποιηθεί (εναλλακτικά) */
instrument = ..., /* Προαιρετική παράμετρος
                                για το ζητούμενο όργανο. Τυχόν
                                απουσία της απαγορεύει τη χρήση
                                απευθείας σύνδεσης */
service = ..., /* Προαιρετική παράμετρος με τιμή
                                κάποιο Φορέα, πχ. 'ΕΜΠ', 'ΥΠΓΕ'.
                                Καθορίζει τον κόμβο του Φορέα στον
                                οποίο ζητείται απευθείας σύνδεση.
                                Αν υπάρχει αυτή η παράμετρος,
                                υπερισχύει της instrument */
DDB = ..., /* Προαιρετική παράμετρος με τιμή 0 ή 1.
                                Αν δεν απουσιάζει και έχει τιμή 1 σημαίνει
                                πως δεν πρέπει να γίνει απευθείας σύνδεση
                                */
LDB = ..., /* Προαιρετική παράμετρος με τιμή 0 ή 1.
                                Αν δεν απουσιάζει και έχει τιμή 1 σημαίνει
                                πως δεν πρέπει να αλλάξει η ενεργή σύνδεση
                                από TBA σε KBA. Αλλιώς παίρνει τιμή ίση με
                                τη μεταβλητή GlobalData.LocalDBOnly που με
                                τη σειρά της παίρνει τιμή από την επιλογή
                                "Μόνο τοπική βάση" στο κύριο μενού */
NoAcct = ... /* Προαιρετική παράμετρος με τιμή 0 ή 1.
                                Αν δεν απουσιάζει και έχει τιμή 1 σημαίνει
                                πως δεν πρέπει να γίνει χρέωση */
);

```

Η τιμή r που επιστρέφει η BeforeQuery () μπορεί να είναι:

- 2. Δεν άλλαξε η ενεργή σύνδεση και χρησιμοποιείται η TBA
- 3. Αλλάξε η ενεργή σύνδεση και χρησιμοποιείται η KBA χωρίς απευθείας σύνδεση
- 4. Όπως το 3 αλλά ταυτόχρονα διαπιστώθηκε πως κάποιος κόμβος ήταν εκτός λειτουργίας. Για το λόγο αυτό η συνάρτηση ReCreateDDBView() επαναδημιούργησε κατά την εκτέλεση της την όλη της KBA με όσους κόμβους "απέμειναν".
- -4. Η παραπάνω διαδικασία επιχειρήθηκε και απέτυχε.
- > 10. Αλλάξε η ενεργή σύνδεση και χρησιμοποιείται η KBA με ενεργή σύνδεση με τον κόμβο r - 10 από τον πίνακα stationids.
- < 0. Αποτυχία.

Αυτό λοιπόν που κάνει η BeforeQuery () είναι:

```

/* Εναρξη χρέωσης και επιστροφή "τρέχουσας" ώρας */
GlobalData.AcctSec = callproc _acct(
    _what = 'start', ...);
/* Αλλαγή ενεργής σύνδεσης, αν ο χρήστης δεν έχει ζητήσει "Μόνο
τοπική βάση" */
if GlobalData.LocalDBOnly != TRUE and LDB != 1 then
    frame.DBSession = GlobalData.DistrDBSession;
else
    return 2;

```

```

endif;
/* Επιστροφή αν δεν πρόκειται να γίνει απευθείας σύνδεση */
if DDB = 1 then
    return 3;
endif;
/* Απευθείας σύνδεση */
... /* Εύρεση κόμβου απευθείας σύνδεσης */
direct connect ...;
return ...;

```

Σημαντική παρατήρηση είναι η ακόλουθη: αν αποτύχει η απευθείας σύνδεση μέσα από την `BeforeQuery()` τότε ο ζητούμενος κόμβος βρίσκεται εκτός λειτουργίας και ούτε η πρόσβαση από την ΚΒΔ θα είναι έτσι και αλλιώς δυνατό να ανακτήσει τα δεδομένα. Για το λόγο αυτό, αν αποτύχει η απευθείας σύνδεση τότε πρέπει ο χρήστης να ενημερωθεί πως τα δεδομένα του δεν είναι προσπελάσιμα και να επιστραφεί κωδικός σφάλματος και όχι να δοκιμαστεί νέα πρόσβαση μέσα από την ΚΒΔ. Αυτά βέβαια σε περίπτωση που δεν έχει από την αρχή ζητηθεί πρόσβαση χωρίς απευθείας σύνδεση (πχ. μέσω της παραμέτρου `DDB` ή με τιμή < 1 στην παράμετρο `instrument`). Σε κάθε περίπτωση, η τελευταία πρακτική πρέπει να αποφεύγεται και να χρησιμοποιείται μόνο σε περίπτωση πραγματικής αδυναμίας χρήσης απευθείας σύνδεσης, πχ. όταν υπάρχουν περισσότερα του ενός όργανα.

5.1.2 Συνάρτηση `AfterQuery()`. Η συνάρτηση αυτή τερματίζει την απευθείας σύνδεση, αν έχει γίνει (με χρήση της `direct disconnect`), αλλάζει ενεργή σύνδεση στην ΤΒΔ (`frame.DBSession = GlobalData.LocalDBSession`) και καταγράφει τα στοιχεία της χρέωσης. Η κλήση της είναι απλούστερη από την `BeforeQuery()`:

```

r = callproc AfterQuery(
    frame, proc, NoAcct, /* Αυτές οι παράμετροι όπως
                          στην BeforeQuery */
    n, level, distr, scale, type,
    instr, display, special /* Αυτές οι παράμετροι όπως
                              στην _acct() */
);

```

Επιστρέφει πάντα 1.

5.2 Πρόσβαση στα δεδομένα από τις εφαρμογές

Η πρόσβαση στα δεδομένα του ΥΔΡΟΣΚΟΠΙΟΥ είναι μια σύνθετη διαδικασία που εμπλέκει αρκετές παραμέτρους της μορφής του συστήματος. Αυτό διότι λόγω της κατανομημένης φύσης της ΒΔ υπάρχει αυξημένη δυσκολία στην ανάκτηση των πραγματικών δεδομένων, πράγμα που σημαίνει πως πρέπει να γίνεται όσο το δυνατό καλύτερη και μεγαλύτερη χρησιμοποίηση των διαχειριστικών πληροφοριών και να αναβάλλεται η πρόσβαση στην ΚΒΔ και το δίκτυο για όσο το δυνατόν αργότερα, όταν θα έχει διαπιστωθεί ποιά ακριβώς δεδομένα είναι διαθέσιμα και που.

Ο τρόπος πρόσβασης στα δεδομένα θα παρουσιασθεί με βάση ένα παράδειγμα στο οποίο ζητούνται από το χρήστη οι πρωτογενείς ημερήσιες πληροφορίες χρονοσειράς επιπέδου 2 για το χρονικό διάστημα 1/1/1970 ως 1/1/1980 του ανεμόμετρου κάποιου σταθμού. Ο σταθμός επιλέγεται από το χρήστη μέσω του υδρομετεωρολογικού χάρτη, του μητρώου σταθμών ή με άλλο τρόπο. Στην εφαρμογή περνά έτσι ο κωδικός ΥΔΡΟΣΚΟΠΙΟΥ του (πεδίο `stations.station`). Στη συνέχεια επιλέγεται και το προς εξέταση όργανο τύπου ANMMTP (πεδίο `instruments.instrument_t`) από τα όργανα του σταθμού, από τα όργανα δηλαδή των οποίων ο κωδικός ΥΔΡΟΣΚΟΠΙΟΥ του σταθμού (πεδίο `instruments.station`) είναι αυτός που ενδιαφέρει. Ο κωδικός ΥΔΡΟΣΚΟΠΙΟΥ του οργάνου (πεδίο `instruments.instrument`) είναι ο βασικός κωδικός μέσω του οποίου θα αναζητηθούν οι πληροφορίες. Εστω πως αυτός ο κωδικός αποθηκεύεται στη μεταβλητή `instr` (τύπου `integer4`) της εφαρμογής. Η γλώσσα που φαίνεται στο παράδειγμα είναι η INGRES Windows/4GL.

- Πρόσβαση στα χαρακτηριστικά χρονοσειρών. Η πρώτη και κυριότερη ενέργεια που πρέπει να γίνει είναι η πρόσβαση στα χαρακτηριστικά των χρονοσειρών. Αυτή η διαχειριστική πληροφορία είναι τύπου `rpl` και άρα διαδέσιμη σε όλους τους κόμβους. Χωρίς λοιπόν πρόσβαση στο δίκτυο μπορεί να διαπιστωθεί αν η ζητούμενη χρονοσειρά για το προς εξέταση διάστημα έχει δεδομένα και επίσης σε ποιόν πίνακα είναι αυτά αποθηκευμένα. Πράγματι, είναι δυνατό να αποθηκεύονται τμήματα της ίδιας χρονοσειράς σε περισσότερους από έναν πίνακες. Αυτό μπορεί να συμβεί πχ. αν για κάποια χρονική περίοδο η αποθήκευση γινόταν σε μια `std` μορφή και έπειτα άλλαξε σε `1st` μορφή. Στην περίπτωση αυτή θα πρέπει οι παρακάτω ενέργειες να επαναληφθούν για κάθε χρονικό διάστημα στο οποίο υπάρχει αλλαγή πίνακα αποθήκευσης των δεδομένων της χρονοσειράς. Στη γενική ωστόσο περίπτωση κατά την οποία ο πίνακας παραμένει σταθερός πρέπει να γίνουν τα ακόλουθα:
- Πρόσβαση σε πίνακες επιπλέον δεδομένων. Αν ο χρήστης επιλέξει να έχει πρόσβαση στα επιπλέον δεδομένα που αποθηκεύονται τοπικά, τότε πρέπει αυτό το γεγονός να ληφθεί υπόψη κατά την ανάκτηση των χαρακτηριστικών χρονοσειρών. Στην περίπτωση της κύριας εφαρμογής του ΥΔΡΟΣΚΟΠΙΟΥ, η επιλογή "Τοπικά Αντίγραφα" του κύριου μενού κάνει την τιμή της μεταβλητής `GlobalData.XPrefix` ίση με `'x_'`. Στην περίπτωση αυτή μπορεί να προσπελασθεί μόνο η ΤΒΔ (δεν επιτρέπεται η πρόσβαση σε πίνακες επιπλέον δεδομένων / τοπικών αντιγράφων απομακρυσμένων κόμβων). Εναλλακτικά, μπορεί ο χρήστης να έχει ζητήσει δεδομένα κάποιας συγκεκριμένης υπηρεσίας, διαφορετικής από τον Φορέα - διαχειριστή του οργάνου. Εστω πως ο ζητούμενος Φορέας βρίσκεται στη μεταβλητή του προγράμματος `serv`. Αν δεν ενδιαφέρει η μεταβλητή έχει τιμή "" (κενό) ή `null`.

- Ανάκτηση του ονόματος του πίνακα αποθήκευσης. Εστω πως το όνομα αποθηκεύεται στη μεταβλητή `TableName`. Τότε για το παράδειγμα ανακτάται καταρχήν ο "κανονικός" πίνακας αποθήκευσης και στη συνέχεια κατασκευάζεται το πλήρες όνομα του πίνακα:

```
select      :TablePrefix = table_p, :format = format
from        timeseries
where       timeseries_t = 'ΠΡΩΤΟΓ' and
           time_res_t = 'ΗΜΕΡΗΣΙΑ' and
           level = 2 and
           start_date <= '1/1/1970' and
           instrument = :instr and
           table_p not like 'x %';
TableName = GlobalData.XPrefix + TablePrefix + '_' +
           format;
```

Το βήμα αυτό μπορεί να παραληφθεί στη γενική περίπτωση κατά την οποία είναι γνωστό εκ των προτέρων το όνομα του πίνακα. Πχ. οι στάθμες - παροχές αποθηκεύονται πάντα στους πίνακες `raw_qh_points` και `raw_qh_curves`. Επίσης το πρόθεμα του ονόματος (`table_p`) μπορεί να θεωρηθεί γνωστό: είναι 'raw' για τα πρωτογενή / παράγωγα δεδομένα και 'aggr' για τα δευτερογενή. Σε καμία όμως περίπτωση δεν μπορεί να θεωρηθεί γνωστή - σταθερή για κάθε όργανο η μορφή αποθήκευσης. Αν γίνει αυτό, μειώνεται η λειτουργικότητα της ΒΔ -αφού πρακτικά δεν επιτρέπεται αλλαγή του τρόπου αποθήκευσης- και αυξάνει η πολυπλοκότητα των εφαρμογών -αφού αντί του παραπάνω γενικού τρόπου πρέπει να αντιμετωπισθεί κάθε όργανο ξεχωριστά.

- Αφού ανακτηθεί το όνομα του πίνακα πρέπει να ανακτηθούν και οι χρησιμοποιούμενες ακρίβειες και μονάδες μέτρησης. Και σε αυτή την περίπτωση ισχύει η ανάγκη χρησιμοποίησης της γενικής μεθόδου και όχι ενσωματωμένης σε ξεχωριστό τμήμα κώδικα για κάθε όργανο λογικής, για τους λόγους που αναφέρθηκαν αμέσως παραπάνω για τη μορφή αποθήκευσης. Επίσης, οι ακρίβειες ή οι μονάδες μέτρησης είναι δυνατό να αλλάζουν ή να ισχύει ειδικός κανόνας για κάποιο Φορέα διαχειριστή (πχ. ταχύτητα ανέμου υποχρεωτικά σε Beaufort) παρά την απαίτηση του ΥΔΡΟΣΚΟΠΙΟΥ για κοινές μονάδες μέτρησης. Σε αυτή την περίπτωση θα πρέπει να αλλάζουν οι εφαρμογές, αν οι μονάδες / ακρίβεια είναι τμήμα τους.

Στην περίπτωση του ανεμόμετρου, στον πίνακα `precision_unit` υπάρχουν δύο εγγραφές για το όργανο με `instrument_t = 'ANMMTP'` και μορφή αποθήκευσης `format = 'std11'` (από την προηγούμενη διαδικασία).

ANMMTP, std11, 0, 0, -1, Μοίρες

ANMMTP, std11, 0, 1, 1, m/sec

Αυτό σημαίνει πως για αυτό το όργανο και αυτή τη μορφή αποθήκευσης, που είναι η πρώτη (πεδίο num = 0), το πρώτο πεδίο value0 (seq = 0) έχει ακρίβεια -1 και μονάδα Μοίρες. Το δε δεύτερο πεδίο value1 (seq = 1) έχει ακρίβεια 1 και μονάδα m/sec. Υπάρχει η PrecisionUnitClass με πεδία precision (smallint) και unit (varchar(16)). Τότε, αν στην εφαρμογή υπάρχει μεταβλητή

```
p = array of PrecisionUnitClass
```

θα είναι:

```
i = 1;
select   :p[i].precision = precision, :p[i].unit = unit
from     precision_units
where    instrument_t = 'ANMMTP' and
         format = :format
order by seq
{
    i = i + 1;
};
```

Μετά το τέλος του select loop στον πίνακα p υπάρχουν οι ακρίβειες και οι μονάδες για όλα τα πεδία value0 ... του πίνακα TableName.

- Αν τα αποτελέσματα τοποθετηθούν στο TableField tf με πεδία

```
varchar(19) DateFld
integer4 StatusFld
float4 Value0Fld
float4 Value1Fld
```

τότε η ανάκτηση τους μπορεί να γίνει με κατάλληλη ερώτηση SQL select στη ΒΔ αφού πριν κληθεί η BeforeQuery() και μετά η AfterQuery() και βέβαια διαιρώντας με την αντίστοιχη ακρίβεια p[].precision. Χρειάζεται ωστόσο προσοχή στο εξής: η ημερομηνία είναι τύπου varchar αλλά στη ΒΔ αποθηκεύεται ως τύπου hdate που είναι στην πραγματικότητα integer4. Αν η ανάκτηση γίνει κατευθείαν σε πεδίο τύπου varchar χάνεται το βασικό πλεονέκτημα της χρήσης hdate που είναι η οικονομία μεταφερόμενων δεδομένων πάνω από το δίκτυο (βλ. παρακάτω) διότι θα μεταφερθούν τα 19 Bytes του πεδίου DateFld και όχι τα 4 Bytes του πεδίου date της ΒΔ. Για το λόγο αυτό πρέπει μαζί με το DateFld να υπάρχει και ένα "κρυφό" πεδίο

```
integer4 IntDateFld
```

στο οποίο αποθηκεύεται απευθείας η τιμή του πεδίου date από τη ΒΔ και μετά, με κατάλληλη μετατροπή βρίσκεται η αντιστοιχία με την τύπου varchar ημερομηνία, καλώντας τη συνάρτηση 3GL IntToRec() μέσα στο select loop της ανάκτησης των

δεδομένων. Επίσης πρέπει να υπάρχει και "κρυφό" πεδίο τύπου `integer4` για κάθε πεδίο δεδομένων, δηλαδή στην προκειμένη περίπτωση:

```
integer4 IntValue0Fld
```

```
integer4 IntValue1Fld
```

ώστε να μεταφέρονται πάνω από το δίκτυο όσα Bytes είναι αποθηκευμένα (1, 2 ή στη χειρότερη περίπτωση 4) και όχι το αποτέλεσμα της διαίρεσης (πάντα 4 Bytes). Επίσης και το πεδίο `StatusFld` πρέπει να είναι κρυφό και να παρουσιάζονται στο χρήστη οι κατάλληλες "σημαίες". Τέλος, οι μονάδες μέτρησης που ανακτήθηκαν στο `p[]` μπορούν να τοποθετηθούν ως τίτλοι του `tf` ή σε άλλο σημείο. Ετσι η ανάκτηση έχει ως εξής:

```
r = callproc BeforeQuery(
    instrument = :instr, /* Το όργανο του οποίου τα
                          δεδομένα ανακτώνται. Αν
                          είναι δυνατό θα γίνει
                          απευθείας σύνδεση */
    service = :serv, /* Η υπηρεσία με την οποία
                      ζητείται απευθείας σύνδεση
                      */
    ...);
if r < 0 then
    callproc AfterQuery();
    callproc MessagePopup(...);
    resume; /* Αποτυχία */
endif
i = 1;
select :tf[i].IntDateFld = int4(date),
       :tf[i].StatusFld = status,
       :tf[i].IntValue0Fld = value0, /* Μεταφέρεται
                                     1 Byte */
       :tf[i].IntValue1Fld = value1 /* Μεταφέρεται
                                     1 Byte */
from :TableName
where date >= '1/1/1970' and date < '1/1/1980'
      /* Η με άλλα κριτήρια επιλογής, ακόμα και
      δυναμικά (πχ. where = :WhereClause) */
order by 1 /* Ταξινόμηση ως προς την ημερομηνία */
{
    callproc IntToRec(
        tf[i].IntDateFld,
        0, 0, 0, 0, 0,
        byref(tf[i].DateFld)); /* int -> varchar */
    tf[i].Value0Fld =
        tf[i].IntValue0Fld / p[0].precision;
    tf[i].Value1Fld =
        tf[i].IntValue1Fld / p[1].precision;
    i = i + 1;
};
callproc SQLStatus(); /* Για commit κτλ. */
callproc AfterQuery(... /* Στοιχεία για χρέωση */);
```

5.3 Διαχειριστικές πληροφορίες

Οι διαχειριστικές πληροφορίες εισάγονται, τροποποιούνται και διαγράφονται μόνο με τη χρήση κατάλληλων ΔΒΔ { Γ } που είναι οι `_station()`, `_instrument()`, `_timeseries()`, `_constant()`, `_event()` αντίστοιχα για σταθμούς, όργανα, χρονοσειρές, σταθερές και γεγονότα. Αυτό γίνεται βέβαια μόνο από τους κωδικούς που ανήκουν στις ομάδες `hadmininsert` (για εισαγωγή) ή `hadminupdate` ή `hdba` (για όλες τις ενέργειες). Οι ΔΒΔ έχουν κοινή μορφή κλήσης από τις εφαρμογές ή από την SQL που είναι η ακόλουθη (ακολουθείται σαν παράδειγμα η κλήση της `_instrument()` με αντίστοιχη κλήση και για τις άλλες):

```
callproc _instrument(
  _table = 'instrument_rpl', /* Το όνομα του πίνακα στον
                             οποίο θα γίνει η ενέργεια.
                             Μπορεί να είναι και
                             'sinstruments_distr' */
  _action = 'insert',      /* Τι θα γίνει. Μπορεί να
                             είναι 'insert', 'update',
                             'delete' */
  instrument = ...,        /* Κωδικός ΥΔΡΟΣΚΟΠΙΟΥ */
  station = ...,
  service = ...,
  ...
                             /* Τα υπόλοιπα πεδία του
                             πίνακα _table */
);
```

Η `_timeseries()` πρέπει να καλείται μετά από κάθε εισαγωγή / τροποποίηση / διαγραφή δεδομένων χρονοσειρών στην ΤΒΔ. Έχει και ένα επιπλέον όρισμα, το `_upd`, το οποίο πρέπει να έχει τιμή 1 (TRUE) για να ενεργοποιείται (μέσω Γεγονότων ΒΔ { Γ } [INGRES91d]) ο δαίμονας ενημέρωσης χρονοσειρών και να τροποποιούνται κατάλληλα οι εγγραφές χαρακτηριστικών χρονοσειρών. Η `_constant()` πρέπει να καλείται μετά από κάθε εισαγωγή / τροποποίηση / διαγραφή σταθερών από την ΤΒΔ. Αντίστοιχα πρέπει να καλείται η `_event()` μετά από κάθε γεγονός σχετικό με τη χρονολογική εξέλιξη των υπόλοιπων διαχειριστικών πληροφοριών. Τα γεγονότα αυτά είναι:

Γεγονός	Πλήρες όνομα γεγονότος	Αντικείμενο γεγονότος
A-ΚΑΤΑΣΚ	ΑΛΛΑΓΗ ΚΑΤΑΣΚΕΥΑΣΤΗ	ΣΤΑΘΜΟΣ
A-ΠΑΡΑΤ	ΑΛΛΑΓΗ ΠΑΡΑΤΗΡΗΤΗ	ΣΤΑΘΜΟΣ
ΕΝΑΡΞΗ	ΕΝΑΡΞΗ	
ΕΛΕΓΧ-Χ	ΕΛΕΓΧΟΣ Χ	ΧΡΟΝΟΣ

ΔΗΜΙΟΥΡΓ	ΔΗΜΙΟΥΡΓΙΑ	ΧΡΟΝΟΣ
ΕΠΕΞΕΡΓ	ΕΠΕΞΕΡΓΑΣΙΑ	ΧΡΟΝΟΣ
ΕΠΙΘΕΩΡ	ΕΠΙΘΕΩΡΗΣΗ	ΣΤΑΘΜΟΣ
ΕΠΙΣΚΕΥΗ	ΕΠΙΣΚΕΥΗ	ΟΡΓΑΝΟ
Α-ΧΡΗΣΤ	ΑΛΛΑΓΗ ΧΡΗΣΤΗ	ΣΤΑΘΜΟΣ
ΔΙΑΚΟΠΗ	ΔΙΑΚΟΠΗ	
ΚΑΤΑΡΓ	ΚΑΤΑΡΓΗΣΗ	ΧΡΟΝΟΣ
ΕΚΤΙΜΗΣΗ	ΕΚΤΙΜΗΣΗ ΕΛΛ. ΤΙΜ.	ΧΡΟΝΟΣ
Α-ΒΗΜΑ	ΑΛΛΑΓΗ ΒΗΜΑΤΟΣ ΜΕΤΡ.	ΟΡΓΑΝΟ
ΕΛΕΓΧ-С	ΕΛΕΓΧΟΣ С	ΧΡΟΝΟΣ
ΕΛΕΓΧ-R	ΕΛΕΓΧΟΣ R	ΧΡΟΝΟΣ
ΟΜΟΓΕΝ	ΟΜΟΓΕΝΟΠΟΙΗΣΗ	ΧΡΟΝΟΣ
ΠΡΟΣΘΗΚΗ	ΠΡΟΣΘΗΚΗ ΔΕΔΟΜ.	ΧΡΟΝΟΣ
Α-ΘΕΣΗΣ	ΑΛΛΑΓΗ ΘΕΣΗΣ	
Α-ΓΕΩΛΟΓ	ΑΛΛΑΓΗ ΓΕΩΛΟΓΟΥ	ΣΤΑΘΜΟΣ
Α-ΥΠΗΡΕΣ	ΑΛΛΑΓΗ ΥΠΗΡΕΣΙΑΣ	ΣΤΑΘΜΟΣ
Ε-ΚΑΤΑΣΚ	ΕΝΑΡΞΗ ΚΑΤΑΣΚΕΥΗΣ	ΣΤΑΘΜΟΣ
Α-ΤΥΠΟΥ	ΑΛΛΑΓΗ ΤΥΠΟΥ	
ΝΕΟ-ΟΡΓ	ΠΡΟΣΘΗΚΗ ΟΡΓΑΝΟΥ	ΣΤΑΘΜΟΣ
ΑΝΑΘΕΩΡ	ΑΝΑΘΕΩΡΗΣΗ	ΧΡΟΝΟΣ
ΣΥΜΠΛ	ΣΥΜΠΛΗΡΩΣΗ	ΧΡΟΝΟΣ
ΕΙΣΑΓΩΓΗ	ΕΙΣΑΓΩΓΗ ΔΕΔΟΜ.	ΧΡΟΝΟΣ
ΕΛΕΓΧ-Т	ΕΛΕΓΧΟΣ Т	ΧΡΟΝΟΣ
ΤΡΟΠΟΠ	ΤΡΟΠΟΠΟΙΗΣΗ	ΧΡΟΝΟΣ
Κ-ΑΝΑΘΕΩ	ΑΝΑΘΕΩΡ. ΚΑΤ. ΕΠ.	ΧΡΟΝΟΣ
Т-ΚΑΤΑΣΚ	ΤΕΛΟΣ ΚΑΤΑΣΚΕΥΗΣ	ΣΤΑΘΜΟΣ

Πίνακας 5.3.1: Οι τύποι των γεγονότων διαχειριστικών πληροφοριών

6 ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΟΡΙΖΟΜΕΝΟΙ ΑΠΟ ΤΟ ΧΡΗΣΤΗ

6.1 Εισαγωγή

Για την ικανοποίηση των διαφόρων ειδικών αναγκών αποθήκευσης και χειρισμού δεδομένων δημιουργήθηκαν διάφοροι νέοι οριζόμενοι από το χρήστη τύποι δεδομένων { Γ } και συναρτήσεις [INGRES91i]. Οι ανάγκες αυτές ήταν είτε η οικονομία σε αποθηκευτικό χώρο και σε κυκλοφορία πάνω από το δίκτυο (πχ. hdate) είτε η ενσωμάτωση νέας λειτουργικότητας (πχ. coord, ordpair). Η ανάπτυξη έγινε σε γλώσσα C και το αποτέλεσμα συνδέθηκε με τον εξυπηρετητή.

Οι νέοι τύποι δεν είναι αντιληπτοί εκτός του εξυπηρετητή των ΤΒΔ. Ο τελευταίος τους αποθηκεύει και τους χειρίζεται κανονικά και ισοδύναμα με τους υπάρχοντες στην εσωτερική τους αναπαράσταση { Γ }, κοκ. Ωστόσο, οι διάφοροι πελάτες (πχ. Windows/4GL, SQL κτλ.) καθώς και ο εξυπηρετητής της ΚΒΔ (STAR) -στην παρούσα τους έκδοση τουλάχιστον- τους αντιλαμβάνονται στην εξωτερική τους αναπαράσταση { Γ }.

Οι νέες συναρτήσεις είναι προσπελάσιμες εκτός του εξυπηρετητή με χρήση των εντολών της SQL, είτε με απευθείας χρήση SQL είτε μέσα από άλλες γλώσσες -Windows/4GL, Embedded SQL/C [INGRES91h] κτλ. Πχ. αν $y = f(x)$ νέα συνάρτηση, `select y = f(x)` είναι νόμιμη χρήση της.

6.2 Τύπος δεδομένων "Συντεταγμένη"

Ο τύπος `coord` αποθηκεύει και χειρίζεται συντεταγμένες της μορφής `[+-]ddd[.mm[.ss]]` με `ddd`: μοίρες δηλαδή $179 \leq ddd \leq 179$, `mm`: πρώτα λεπτά, δηλαδή $0 \leq mm \leq 59$ και `ss`: δευτερόλεπτα, δηλαδή $0 \leq ss \leq 59$.

Η εσωτερική αναπαράσταση { Γ } είναι `integer4` με τιμή $ddd * 3600 + mm * 60 + ss$ δηλαδή το συνολικό αριθμό δευτερολέπτων της συντεταγμένης. Πχ. η συντεταγμένη '10.30.21' αναπαρίσταται εσωτερικά ως ένας ακέραιος 4 Bytes με τιμή $10 * 3600 + 30 * 60 + 21 = 37821$.

Αντίστοιχα, η εξωτερική αναπαράσταση στους πελάτες είναι `char(10)` με την παραπάνω μορφή. Πχ. '34.56.1', '123', '15.37.47', '-78.42', '-78.' κτλ.

Οι ορισμένες μετατροπές { Γ } από και προς άλλους τύπους είναι οι ακόλουθες: `coord <-> int1`, `coord <-> int2`, `coord <-> int4`, `coord <-> float4`, `coord <-> float8`, `coord <-> char`, `coord <-> varchar`. Οι μετατροπές αυτές συμβαίνουν τόσο με χρήση των κατάλληλων συναρτήσεων `int1()`, `int2()` κτλ., όσο και αυτόματα, όταν αυτό χρειαστεί, δηλαδή σε εκτέλεση πράξεων, συγκρίσεων κοκ. Πχ. αν `x` είναι `coord`

έχουν νόημα οι ακόλουθες εκφράσεις: `int4(x)`, `float4(x)`, `varchar(x)`, `coord('34.56.12')` κτλ.

Οι ορισμένοι τελεστές είναι οι ακόλουθοι: `!=`, `=`, `>`, `>=`, `<`, `<=`. Στην περίπτωση αυτή η μετατροπή τύπου γίνεται αυτόματα. Πχ. αν `x` είναι `coord` οι ακόλουθες εκφράσεις έχουν νόημα: `x = '34.56.1'`, `x < '-80.32'`, `x != 36000` (σημειωτέον πως 36000 δευτερόλεπτα = '10.00.00'), `x >= 72000.0` κτλ.

Οι ορισμένες πράξεις είναι οι: `+` και `-` με αυτόματη μετατροπή τύπου. Πχ. αν `x` είναι `coord` έχουν νόημα οι ακόλουθες εκφράσεις: `x = '28.00.00' + '36.21'`, `x = 36000 - 3600.0` κτλ.

Τέλος, υπάρχει η ορισμένη συνάρτηση

`coord(x)`

για μετατροπή μεταβλητής `x` άλλου τύπου σε `coord`. Πχ. `coord(int4(x))`, `coord('34.56.1')`, `coord(36060)` κτλ.

6.3 Τύπος δεδομένων "Διατεταγμένο ζεύγος"

Ο τύπος `ordpair` αποθηκεύει και χειρίζεται διατεταγμένα ζεύγη ή χωρικά σημεία της μορφής `(a, b)`.

Εσωτερικά αναπαριστάται ως ζεύγος πραγματικών αριθμών διπλής ακρίβειας, δηλαδή `float8 (double) a, b`. Οι αριθμοί αυτοί μπορεί να είναι και `integer` (ακέρατοι) ή `coord` (συντεταγμένες), ωστόσο εσωτερικά αποθηκεύονται ως `float8` (πραγματικοί). Για λογικό αριθμό γηφίων (λιγότερα από 16) δεν υπάρχει απώλεια ακρίβειας.

Αντίστοιχα, η εξωτερική αναπαράσταση είναι: `char(26)` με μορφή `'([+-]intpart.fraction, [+]-intpart.fraction)'` όπου `intpart` το ακέραιο μέρος με μήκος 8 χαρακτήρες και `fraction` το δεκαδικό με μήκος 3 χαρακτήρες. Πχ. `'(10567.340, -456.920)'`, `'(12345678.901, -98765432.109)'` κτλ.

Οι ορισμένες μετατροπές (`coercions`) από και προς άλλους τύπους είναι οι: `ordpair <-> char` και `ordpair <-> varchar`. Οι μετατροπές αυτές συμβαίνουν τόσο με χρήση των κατάλληλων συναρτήσεων `char()`, `varchar()`, όσο και αυτόματα, όταν αυτό χρειαστεί. Πχ. αν `x` είναι `ordpair` οι ακόλουθες εκφράσεις έχουν νόημα: `char(x)`, `varchar(x)` κτλ.

Οι ορισμένοι τελεστές είναι βέβαια οι: `!=`, `=`, `>`, `>=`, `<`, `<=` με αυτόματη μετατροπή τύπου. Πχ. αν `x` είναι `ordpair` οι ακόλουθες εκφράσεις έχουν νόημα: `x = '(123, 555.02)'`, `x < '(80.32, 45)'`, `x != '(1, 0)'` κτλ. Η ανισότητα ορίζεται με σύγκριση καταρχήν των "πρώτων" στοιχείων κάθε ζεύγους και στη συνέχεια των δεύτερων.

Οι ορισμένες πράξεις είναι οι + και - με αυτόματη μετατροπή τύπου έτσι ώστε αν x είναι ordpair οι ακόλουθες εκφράσεις να έχουν νόημα: $x = '(158.00, 34)'$ + $'(3, 6.21)'$ κτλ.

Τέλος, οι ορισμένες συναρτήσεις είναι οι ακόλουθες:

`ordpair(x, y)`

για δημιουργία διατεταγμένου ζεύγους (ordpair) από τα δυο ορίσματα x, y (ίδιου τύπου integer, float ή coord.

Πχ. `ordpair(coord('34.56.1'), coord(36060)), ordpair('56.78', varchar(12.1)), ordpair(48.578, float8(0))` κτλ.

`distance(x, y)`

με ορίσματα δύο διατεταγμένα ζεύγη x, y για υπολογισμό της απόστασης τους (float8).

Πχ. `distance(ordpair(coord('45.16'), coord(0)), ordpair(0, 5.3)), distance('(34.59, 98.782)', '(333, 12.01)')` κτλ.

`direction(x, y)`

για υπολογισμό της κλίσης (float8) ενός ευθύγραμμου τμήματος με άκρα τα διατεταγμένα ζεύγη x, y.

Πχ. `direction(ordpair(3, 4), ordpair(coord(0), coord(90000)))` κτλ.

`midpoint(x, y)`

για υπολογισμό του μέσου (ordpair) του ευθύγραμμου τμήματος με άκρα τα διατεταγμένα ζεύγη x, y.

Πχ. `midpoint(ordpair(57.04, float8('36.632')))`

`apart(x)`

επιστρέφει (float8) το πρώτο στοιχείο του διατεταγμένου ζεύγους x

Πχ. `apart(ordpair(coord('43.27'), coord('-110.32.49')))`

`bpart(x)`

επιστρέφει (float8) το δεύτερο στοιχείο του διατεταγμένου ζεύγους x

Πχ. `bpart('(146.8, 34.02)')`

`philambda2xy(z)`

επιστρέφει διατεταγμένο ζεύγος (`ordpair`) με τιμές τις επίπεδες συντεταγμένες x και y του διατεταγμένου ζεύγους z του οποίου το πρώτο στοιχείο είναι γεωγραφικό πλάτος (φ) και το δεύτερο γεωγραφικό μήκος (λ) δηλαδή $z = (\varphi, \lambda)$

Πχ. `apart(philamda2xy(ordpair(coord('36.18'), coord(100000))))`, `apart(philamda2xy('(12345.67, 0.45)'))`

`xy2philamda(z)`

είναι η αντίστροφη συνάρτηση της `philamda2xy()` και επιστρέφει διατεταγμένο ζεύγος (`ordpair`) με τιμές τα φ και λ διατεταγμένου ζεύγους z με επίπεδες συντεταγμένες x και y .

Πχ. `xy2philamda(philamda2xy(z))`,
`bpart(xy2philamda(ordpair(coord(0), coord(0))))`

6.4 Τύπος δεδομένων "Ημερομηνία ΥΔΡΟΣΚΟΠΙΟΥ"

Ο τύπος δεδομένων `hdate` (`HydroscopeDATE`) αποθηκεύει και χειρίζεται ημερομηνίες με ακρίβεια πρώτου λεπτού, ώστε να μην απαιτείται η χρήση του ενδοκτισμένου τύπου `date` που καταλαμβάνει 12 bytes. Δεν απαιτείται μεγαλύτερη ακρίβεια (πχ. δευτερόλεπτα) για τις εφαρμογές του ΥΔΡΟΣΚΟΠΙΟΥ. Η μέγιστη λοιπόν χρονική διακριτότητα στο ΥΔΡΟΣΚΟΠΙΟ είναι 1 πρώτο λεπτό.

Η εσωτερική αναπαράσταση είναι `integer4` με τιμή τον αριθμό των πρώτων λεπτών από την ώρα 00:00 της 1/1/1. Με βάση αυτό, καλύπτεται η αποθήκευση περίπου 4085 ετών πριν και μετά από την παραπάνω ημερομηνία για θετικές ή αρνητικές τιμές αντίστοιχα. Το εύρος αυτό θεωρείται αρκετό.

Η εξωτερική αναπαράσταση είναι `char(19)` με μορφή: `'dd/mm/yyyy [hh:mm[:00]]'` όπου `dd` η ημέρα, `mm` ο μήνας, `yyyy` το έτος, `hh` η ώρα και `mm` το πρώτο λεπτό. Τα δευτερόλεπτα αγνοούνται στην εισαγωγή και είναι 00 στην εξαγωγή δεδομένων αυτού του τύπου. Επίσης επιτρέπεται η χρήση και των συμβόλων `'.'` και `'-'` εκτός του `'/'` για διαχωρισμό των στοιχείων της ημερομηνίας (αλλά πρέπει το ίδιο σύμβολο να χρησιμοποιηθεί και τις 2 φορές). Πχ. `'17/8/1934 12:52:00'`, `'12.03.1993'`, `'15/08/47 19:10'`, `'06/12/1941 07:00'`, `'7-3-1834 15:45:32'` κτλ.

Οι ορισμένες μετατροπές από και προς άλλους τύπους είναι οι ακόλουθες: `hdate <-> int1`, `hdate <-> int2`, `hdate <-> int4`, `hdate <-> float4`, `hdate <-> float8`, `hdate <-> char`, `hdate <-> varchar`. Οι μετατροπές αυτές συμβαίνουν τόσο με χρήση των κατάλληλων συναρτήσεων `int1()`, `int2()` κτλ, όσο και αυτόματα, όταν αυτό χρειαστεί. Πχ. αν x είναι `hdate` οι ακόλουθες εκφράσεις έχουν

νόημα: `int4(x)`, `float4(x)`, `varchar(x)`, `hdate('12.5.1902 15:03')`,
`hdate(12345678)`, `hdate(1.0)` κτλ.

Οι ορισμένοι τελεστές είναι: `!=`, `=`, `>`, `>=`, `<`, `<=` με αυτόματη μετατροπή τύπου. Πχ. αν `x` είναι `hdate` οι ακόλουθες εκφράσεις έχουν νόημα: `x = '14/9/1'`, `x < '22-1-1993 12:09'`, `x != 12360900`, `x >= 122272000.0` κτλ.

Οι ορισμένες πράξεις είναι: `+` και `-` με αυτόματη μετατροπή τύπου. Πχ. αν `x` είναι `hdate` οι ακόλουθες εκφράσεις έχουν νόημα: `x = '7/9/1928' + '21.12.1990 17:02'`, `x = 1115467 - 45467` κτλ.

Σχετικά με την ανάκτηση `hdate` πάνω από το δίκτυο είναι πολύ σημαντική η ακόλουθη παρατήρηση: ο τύπος αυτός δημιουργήθηκε για οικονομία σε αποθηκευτικό χώρο και σε κυκλοφορία πάνω από το δίκτυο (4 Bytes σε σχέση με τα 12 Bytes του ενδοκτισμένου τύπου `date`). Από τη στιγμή όμως που η εξωτερική του αναπαράσταση είναι `char(19)`, τότε αν το πεδίο `x` της ΒΔ είναι τύπου `hdate`, η ερώτηση SQL από κάποιο απομακρυσμένο κόμβο

```
select x from <table>...
```

θα επιστρέφει πεδία εύρους 19 Bytes, αναιρώντας έτσι το πλεονέκτημα της οικονομίας κυκλοφορίας πάνω από το δίκτυο. Για το λόγο αυτό πρέπει να ανακτάται η ακέραια τιμή του πεδίου και στη συνέχεια να καλείται η συνάρτηση `IntToRec()` για μετατροπή του ακέραιου σε τύπο `hdate`. Δηλαδή:

```
select int4(x) from <table>...
{
    callproc IntToRec(
        tf[i].IntDateFld,
        0, 0, 0, 0, 0,
        byref(tf[i].DateFld));
    ...
};
```

Οι ορισμένες συναρτήσεις είναι οι ακόλουθες:

`hdate(x)`

για μετατροπή μεταβλητής `x` άλλου τύπου σε `hdate`.

Πχ. `hdate(int4(x))`, `hdate('8/8/1888 08:08')`, `hdate(341131.98)`
 κτλ.

`hdate_part(type, x)`

Αν `x` είναι τύπου `hdate` και `type` είναι συμβολοσειρά με τιμή `'year'`, `'month'`, `'day'`, `'hour'` ή `'minute'` τότε επιστρέφεται αντίστοιχα το έτος, ο μήνας, η ημέρα, η ώρα ή το πρώτο λεπτό του `x`. Η συνάρτηση αυτή είναι αντίστοιχη της

`date_part()` της SQL. Πχ. `hdate_part('year', '1/1/1994')`,
`hdate_part('hour', '15/10/1990 13:22')` κτλ.

6.5 Τύπος δεδομένων "Κωδικός ΥΔΡΟΣΚΟΠΙΟΥ"

Ο τύπος δεδομένων `hsiid` (Hydroscope Station & Instrument ID) αποθηκεύει και χειρίζεται κωδικούς οργάνων και σταθμών του ΥΔΡΟΣΚΟΠΙΟΥ. Κάθε σταθμός έχει έναν αρχικό κωδικό που είναι πχ. ένας αύξοντας αριθμός του. Επίσης, κάθε όργανο έχει έναν αρχικό κωδικό που είναι ο αύξων αριθμός του (1ο, 2ο, 3ο κτλ. όργανο) στο σταθμό που ανήκει. Οι δύο κωδικοί συνδυάζονται όπως θα φανεί παρακάτω.

Η εσωτερική αναπαράσταση είναι `integer4` με τιμή την ακόλουθη: αν αποθηκεύεται κωδικός σταθμού, από τα 4 Bytes τα 3 αριστερότερα είναι ο κωδικός του σταθμού και το δεξιότερο είναι 0. Έτσι, ο τελικός κωδικός του σταθμού είναι ο αρχικός κωδικός του μετακινημένος { Γ } 8 bits αριστερά, στην πραγματικότητα δηλαδή πολλαπλασιασμένος¹¹ με 256. Έτσι, ο σταθμός 1 θα έχει κωδικό 256, ο 2 512, ο 10 2560 κοκ. Αν πάλι πρόκειται για όργανο, στα 3 αριστερότερα Bytes αποθηκεύεται ο αρχικός κωδικός του σταθμού και στο δεξιότερο ο αύξων αριθμός του οργάνου σε αυτόν τον σταθμό. Πχ. το όργανο 1 του σταθμού 1 θα έχει κωδικό 257 (= 256 + 1), το όργανο 3 του σταθμού 2 θα έχει κωδικό 515 (= 512 + 3) κοκ. Γενικά ο *i*-οστός σταθμός θα έχει κωδικό $256 * i + 0$ και το *k*-οστό όργανο αυτού του σταθμού θα έχει κωδικό $256 * i + k$. Με αυτό τον τρόπο είναι δυνατή η αποθήκευση περίπου 16,7 εκατομμυρίων σταθμών με ως 127 όργανα ανά σταθμό, αριθμοί που υπερκαλύπτουν τις ανάγκες του Ελληνικού μετρητικού δικτύου.

Η εξωτερική αναπαράσταση είναι `integer4`.

Οι ορισμένες μετατροπές από και προς άλλους τύπους και οι ορισμένοι τελεστές είναι οι ίδιοι με των ενδοκτισμένων τύπων ακεραίων `integer4`.

Οι ορισμένες πράξεις είναι $n +$ και $n -$ με αυτόματη μετατροπή τύπου.

Οι ορισμένες συναρτήσεις είναι οι ακόλουθες:

`hsiid(x)`

για μετατροπή μεταβλητής *x* άλλου τύπου σε `hsiid`, με λειτουργία παρόμοια με τις αντίστοιχες συναρτήσεις των ακεραίων (`int1()`, `int2()` κτλ.).

¹¹ Στη δυαδική λογική μετακίνηση 1 bit αριστερά ισοδυναμεί με πολλαπλασιασμό με το 2, όπως και μετακίνηση 1 bit δεξιά ισοδυναμεί με διαίρεση με το 2. Αυτός είναι και ο γρηγορότερος τρόπος πολλαπλασιασμού / διαίρεσης σε περίπτωση που ο πολλαπλασιαστής / διαιρέτης είναι δύναμη του 2: αριστερή / δεξιά μετακίνηση τόσα bits όσα είναι η δύναμη του 2 του αριθμού. Συνεπώς, αριστερή μετακίνηση 8 bits ισοδυναμεί με πολλαπλασιασμό με $2^8 = 256$.

Πχ. `hsiid(257)`, `hsiid('2567')`, `hsiid(25600.98)` κτλ.

`spart(x)`

επιστρέφει τον (αρχικό) κωδικό - αύξοντα αριθμό σταθμού (`integer4`) ενός κωδικού ΥΔΡΟΣΚΟΠΙΟΥ `x`

Πχ. `spart(256)`, `spart(514)`, `spart(int4(z))`, `spart(3245.67)` κτλ.

`ipart(x)`

επιστρέφει τον (αρχικό) κωδικό - αύξοντα αριθμό οργάνου (`integer4`) ενός κωδικού ΥΔΡΟΣΚΟΠΙΟΥ `x`

Πχ. `ipart('563412')`, `ipart(3)` κτλ.

`station2hsiid(x)`

δοθέντος του αρχικού κωδικού - αύξοντος αριθμού `x` ενός σταθμού, επιστρέφει (`hsiid`) τον τελικό κωδικό του

Πχ. `station2hsiid(17)`, `station2hsiid(145.67)` κτλ.

`instrument2hsiid(x, y)`

δοθέντος του αρχικού κωδικού - αύξοντος αριθμού `y` ενός οργάνου και του αρχικού κωδικού - αύξοντος αριθμού `x` του σταθμού στον οποίο αυτό ανήκει, επιστρέφει (`hsiid`) τον τελικό κωδικό του

Πχ. `instrument2hsiid(14, 8)`, `instrument2hsiid(int4('51'), 2)`, `instrument2hsiid(spart(512), ipart(513))` κτλ.

6.6 Περιορισμοί χρήσης συναρτήσεων οριζόμενων από το χρήστη

Οι οριζόμενες από το χρήστη συναρτήσεις μπορούν να χρησιμοποιηθούν σε τοπικό επίπεδο, δηλαδή σε μη κατανεμημένη ΒΔ, τοπικά σε κάθε κόμβο, από τον εξυπηρετητή και όλους τους πελάτες (SQL, Windows/4GL κοκ) ισοδύναμα με τις ενδοκτισμένες συναρτήσεις. Έτσι, στην ΤΒΔ η `int4()` και η `hsiid()` είναι πρακτικά ισοδύναμες. Ωστόσο αυτό δεν ισχύει στη λειτουργία της ΚΒΔ. Στην περίπτωση αυτή, επειδή ο εξυπηρετητής της ΚΒΔ (STAR) δεν έχει τη δυνατότητα αναγνώρισης και χειρισμού των νέων συναρτήσεων, δεν είναι δυνατό να χρησιμοποιούνται αυτές οι συναρτήσεις σε κατανεμημένη λειτουργία. Δηλαδή η ερώτηση SQL

```
select int4(hdate('3/12/1990'))
```

είναι νόμιμη στην ΤΒΔ αλλά όχι στην ΚΒΔ.

Εκ πρώτης όψεως αυτός ο περιορισμός φαίνεται πολύ σοβαρός, ότι σημαίνει πρακτικά πως πρέπει να υπάρχουν δυο τμήματα κώδικα όταν χρησιμοποιούνται αυτές οι συναρτήσεις, ένα για την ΤΒΔ και ένα για την ΚΒΔ. Ωστόσο τελικά αυτό δεν συμβαίνει διότι:

- Το μεγαλύτερο τμήμα του κώδικα λειτουργεί στη σύνδεση με την ΤΒΔ. Έτσι, όλες οι προσβάσεις σε διαχειριστικές πληροφορίες και σε πληροφορίες εφαρμογών, καθώς και όλες οι προσβάσεις σε δεδομένα για εγγραφή γίνονται σε τοπικό επίπεδο (δεν επιτρέπεται τροποποίηση δεδομένων απομακρυσμένου κόμβου). Έτσι, η μόνη περίπτωση που χρησιμοποιείται η σύνδεση με την ΚΒΔ είναι οι περιπτώσεις ανάκτησης δεδομένων σταθερών και χρονοσειρών και σπανιότερα η ανάκτηση διαχειριστικών πληροφοριών για δευτερεύοντες σταθμούς.
- Για λόγους ταχύτητας δεν χρησιμοποιείται η κανονική πρόσβαση πάνω από την ΚΒΔ αλλά μέσα από την `BeforeQuery()` γίνεται στη μεγάλη πλειοψηφία των περιπτώσεων απευθείας σύνδεση { Γ } με την ΤΒΔ που περιέχει τις ζητούμενες πληροφορίες. Στην περίπτωση αυτή μπορούν να χρησιμοποιηθούν οι οριζόμενες από το χρήστη συναρτήσεις χωρίς προβλήματα. Σε περίπτωση αποτυχίας της απευθείας σύνδεσης έτσι και αλλιώς τα δεδομένα δεν είναι προσπελάσιμα.
- οι μόνες περιπτώσεις που πρέπει να γίνει ειδική πρόβλεψη για τη λειτουργία των από το χρήστη οριζόμενων συναρτήσεων είναι όταν υπάρχει λόγος πρόσβασης χωρίς απευθείας σύνδεση στην ΚΒΔ. Στην περίπτωση αυτή η χρήση των συναρτήσεων θα πρέπει να γίνεται με προσοχή ή να χρησιμοποιούνται εναλλακτικές λύσεις.

6.7 Συνάρτηση ελέγχου τύπου

Η συνάρτηση `checktype(type, entry)` ελέγχει αν το όρισμα `entry` είναι του τύπου που περιγράφεται στο όρισμα `type`. Και τα δύο ορίσματα είναι συμβολοσειρές. Το δεύτερο είναι αυτό που θα ελεγχθεί. Το πρώτο μπορεί να παίρνει τιμές 'integer', 'float', 'char', 'varchar', 'coord', 'hdate', 'date', 'hsiid' κοκ. Η συνάρτηση επιστρέφει -1 αν το δεύτερο όρισμα δεν είναι του τύπου που περιγράφεται στο πρώτο όρισμα. Διαφορετικά επιστρέφει το πραγματικό μήκος (μεγαλύτερο από το 0), αν εξαιρεθούν τα κενά στην αρχή και το τέλος της συμβολοσειράς, του δεύτερου ορίσματος. Η συνάρτηση μπορεί να χρησιμοποιηθεί σε κάθε περίπτωση που είναι επιθυμητός ο έλεγχος αν κάποια συμβολοσειρά μπορεί να αναπαραστήσει μια μεταβλητή του ζητούμενου τύπου. Κυρίως βέβαια σε περιπτώσεις που ζητείται από το χρήστη της εφαρμογής η εισαγωγή ενός συγκεκριμένου τύπου (πχ. ενός ακεραίου ή κινητής υποδιαστολής ή ημερομηνίας) και ζητείται να ελεγχθεί αν ο χρήστης όντως εισήγαγε τιμή του ζητούμενου τύπου, οπότε καλείται -όπως για όλες τις κλήσεις παρόμοιων από το χρήστη οριζόμενων συναρτήσεων- με ερώτηση SQL προς τον εξυπηρετητή της μορφής


```
select :r = checktype(:type, :entry)
```

όπου *r* η μεταβλητή της γλώσσας όπου θα αποθηκευθεί η απάντηση, *type* ο ζητούμενος τύπος και *entry* η τιμή (συμβολοσειρά) που εισήγαγε ο χρήστης.

6.8 Συναρτήσεις λέξης κατάστασης

Για τη διαχείριση των λέξεων κατάστασης { Γ } έχουν δημιουργηθεί ορισμένες νέες συναρτήσεις, οι `getstatus()`, `setstatus()`, `getstatusall()` και `getstatusdata()`.

Η συνάρτηση `getstatus(status, action)` επιστρέφει την τιμή των πεδίων - σημαίων - που περιγράφονται στη συμβολοσειρά *action* της `integer4` λέξης κατάστασης *status*. Η συμβολοσειρά *action* περιέχει ένα ή περισσότερα από τα χαρακτηριστικά γράμματα των πεδίων της λέξης κατάστασης. Στην απλοποιημένη και πλέον συνηθισμένη και χαρακτηριστική μορφή της, ζητείται η τιμή ενός πεδίου *bits* - σημαίας με ένα μόνο χαρακτηριστικό γράμμα στην παράμετρο *action*. Για παράδειγμα η ερώτηση SQL:

```
select date, value0 from raw_std2
where instrument = ... and getstatus(status, 'L') = 1
```

επιστρέφει τις εγγραφές εκείνες των οποίων το πεδίο L (επίπεδο 1) της λέξης κατάστασης είναι 1, τις εγγραφές εκείνες δηλαδή που ανήκουν στο επίπεδο 1. Αντίστοιχα παραδείγματα είναι τα: `getstatus(status, 'R')` (μπορεί να επιστρέφει τιμές 0, 1, 2, 3 ή 4), `getstatus(status, 'Q')` (μπορεί να επιστρέφει τιμές 0, 1, 2 ή 3), `getstatus(0x30200a1, 'X')` (επιστρέφει 2 που είναι η τιμή του πεδίου X - προσοχή στο ό,τι σε αυτό το παράδειγμα το *status* είναι δεκαεξαδική σταθερά) κ.ο.κ.

Η πλέον σύνθετη αλλά και δύσχρηστη μορφή της `getstatus()` είναι αυτή στην οποία ζητούνται και επιστρέφονται ταυτόχρονα πολλές τιμές πεδίων *bits* της λέξης κατάστασης. Αυτό γίνεται αν στην παράμετρο *action* περιέχονται περισσότερα του ενός πεδία *bits* με αντίστοιχα χαρακτηριστικά γράμματα. Πχ. `getstatus(status, 'LM')` (για το αν η εγγραφή ανήκει στα επίπεδα 1 και 2), `getstatus(status, 'XTQ')` (για τις τιμές των ελέγχων χωρικής και χρονικής συνέπειας και του δείκτη ποιότητας) κ.ο.κ. Στην περίπτωση αυτή η `integer4` (4 Bytes = 32 bits) τιμή που επιστρέφεται αποτελείται και αυτή από πεδία *bits* μήκους το καθένα 4 bits δηλαδή μέγιστο 8 πεδία - άρα και μέχρι 8 χαρακτηριστικά γράμματα. Το πλέον "δεξιά" πεδίο 4 bits της λέξης επιστροφής έχει τιμή την τιμή του πεδίου της λέξης κατάστασης που το αντίστοιχο της γράμμα στην παράμετρο *action* ήταν δεξιότερο. Το αμέσως αριστερότερο γράμμα αντιστοιχεί στο αμέσως αριστερότερο πεδίο 4 bits της λέξης επιστροφής κ.ο.κ. Στα παραπάνω παραδείγματα, αν η λέξη κατάστασης είναι τέτοια που οι τιμές των πεδίων να είναι L = 1, M = 0, X = 2, T = 4 και Q = 1 τότε η

`getstatus(status, 'LM')` επιστρέφει 0x10 (δεκαδικό 16) ενώ η `getstatus(status, 'XTQ')` επιστρέφει 0x241 (δεκαδικό 577). Προφανώς αυτός ο τρόπος είναι αρκετά δύσχρηστος και γι' αυτό το λόγο μπορεί να είναι πρακτικός μόνο σε ειδικές περιπτώσεις. Συνιστάται, αν παρίσταται ανάγκη ανάκτησης ταυτόχρονα πολλών πεδίων κατάστασης, να γίνονται πολλαπλές κλήσεις στην `getstatus()` πχ. `getstatus(status, 'L') = 1 and getstatus(status, 'M') = 0` κοκ.

Η συνάρτηση `setstatus(status, action)` δέχεται ως όρισμα την παλιά λέξη κατάστασης `statusword` -n οποία προφανώς διαβάζεται από το αντίστοιχο πεδίο της ζητούμενης εγγραφής- και μια συμβολοσειρά που περιέχει τις επιθυμητές τιμές των επιμέρους πεδίων της λέξης κατάστασης. Η μορφή της είναι '`<letter><value>` [`<letter><value>`]' όπου `<letter>` είναι το χαρακτηριστικό γράμμα του αντίστοιχου πεδίου (δηλαδή 'L', 'M', 'X' κοκ) και `<value>` ένας αριθμός που είναι η τιμή που θα δοθεί στο πεδίο. Αν υπάρχουν πολλές τέτοιες δυάδες `<letter><value>` τότε όλα τα αντίστοιχα στα γράμματα επιμέρους πεδία της λέξης κατάστασης θα πάρουν τις αντίστοιχες τιμές. Η συνάρτηση επιστρέφει τη νέα λέξη κατάστασης, όπου τα bits των τροποποιούμενων πεδίων έχουν πάρει τις νέες τιμές, μαζί με τα "απέιραχτα" bits της παλιάς λέξης κατάστασης.

Για παράδειγμα, αν αρχικά για κάποια εγγραφή το πεδίο λέξης κατάστασης είναι 0, τότε η `setstatus(status, 'X2Q3F1L1')` θα επιστρέψει 0x302000c1 (που σημαίνει πως F = 1, I = 1, X = 2, Q = 3, L = 1 δηλαδή παρουσιάστηκαν συνθήκες υπόβαθρου παγετού και πάγου, ο έλεγχος χωρικής συνέπειας έγινε και απέτυχε, ο δείκτης ποιότητας της εγγραφής είναι 3 δηλαδή εγγραφή λάθος και η εγγραφή ανήκει στο επίπεδο 1). Αν στη συνέχεια σε αυτό το πεδίο `status` εφαρμοσθεί `setstatus(status, 'T1A1F0M1')` θα επιστραφεί 0x112000d3 (που σημαίνει F = 0, T = 1, A = 1, M = 1 και θα παραμείνουν τα I = 1, Q = 3, L = 1 και X = 2, δηλαδή τελικά παρουσιάστηκαν συνθήκες υπόβαθρου πάγου, ο έλεγχος χωρικής συνέπειας έγινε και απέτυχε, ο έλεγχος χρονικής συνέπειας έγινε και πέτυχε, ο δείκτης ποιότητας είναι 3 δηλαδή εγγραφή λάθος, υπάρχει αλφαριθμητικό σχόλιο για την εγγραφή αυτού του οργάνου σε αυτή την ημερομηνία και η εγγραφή ανήκει στα επίπεδα 1 και 2). Η νέα τιμή της λέξης κατάστασης αποθηκεύεται πλέον στο πεδίο λέξης κατάστασης του πίνακα πρωτογενών με εντολή SQL της μορφής:

```
update ... set status = setstatus(status, 'T1A1F0M1')
```

Σημασία έχει το γεγονός πως οι ακριβείς τιμές της λέξης κατάστασης δεν έχουν σημασία για το χρήστη ή τον προγραμματιστή. Ο τελευταίος δεν χρειάζεται να "αποκρυπτογραφεί" τη λέξη κατάστασης αφού για αυτό το σκοπό υπάρχει η συνάρτηση `getstatus()`. Επίσης, η λέξη κατάστασης διαβάζεται από τη ΒΔ με `select` και γράφεται στη ΒΔ με `insert / update`, χωρίς να χρειάζεται να χρησιμοποιηθεί στο πρόγραμμα. Μόνο το αποτέλεσμα της `getstatus()` χρησιμοποιείται για όποιο πεδίο χρειάζεται.

Η συνάρτηση `getstatusall(status)` επιστρέφει μια συμβολοσειρά που περιέχει τις τιμές των πεδίων - σημαίων που είναι διάφορες από το 0, μαζί με τα χαρακτηριστικά τους γράμματα, σε μορφή αντίστοιχη με το πεδίο `action` της `setstatus()`. Πχ. αν `status = 0x112000d3`, η κλήση `setstatus(status)` επιστρέφει `'L1M1A1Q3X2T1I1'`. Είναι προφανώς πολύ χρήσιμη σε περίπτωση που χρειάζεται παρουσίαση όλων των "ένεργων" (διάφορων από 0) σημαίων της λέξης κατάστασης.

Τέλος, η συνάρτηση `getstatusdata(letter)` δίνει πληροφορίες για το εύρος σε bits και την ακριβή θέση κάθε πεδίου bits - σημαίας της λέξης κατάστασης. Επιστρέφει έναν ακέραιο `integer2 16 bits` του οποίου τα 8 δεξιότερα bits παίρνουν τιμή το εύρος σε bits του αντίστοιχου πεδίου της λέξης κατάστασης και τα 8 αριστερότερα bits τη θέση του πεδίου στη λέξη κατάστασης. Για παράδειγμα, `getstatusdata('L')` επιστρέφει `0x0001` (δεκαδικό 1), `getstatusdata('Q')` επιστρέφει `0x0602` (δεκαδικό 1538) ενώ `getstatusdata('T')` επιστρέφει `0x1803` (δεκαδικό 6147). Η συνάρτηση αυτή χρειάζεται να χρησιμοποιείται μόνο σε περίπτωση που πρέπει να είναι γνωστές οι πληροφορίες αυτές για τα πεδία bits.

7 ΑΝΤΙΚΕΙΜΕΝΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

7.1 Εισαγωγή

Στη συνέχεια παρουσιάζονται αναλυτικά οι πίνακες και τα άλλα αντικείμενα που απαρτίζουν τη βάση δεδομένων του ΥΔΡΟΣΚΟΠΙΟΥ. Το σχήμα της ΤΒΔ και της ΚΒΔ είναι κοινό, γι' αυτό και δεν γίνεται διάκριση ανάμεσα στις δύο¹². Για όλους τους πίνακες όμως η αναφορά του τύπου τους (rpl, rmt, κοκ) καθορίζει ανάλογα και τον τρόπο με τον οποίο σχετίζονται αυτοί οι πίνακες με τους πίνακες της ΚΒΔ. Έτσι, οι πίνακες rpl αντιγράφονται σε όλους τους κόμβους σε αντίστοιχους πίνακες rmt, ενώ η λογική ένωση όλων των πινάκων xxx_rpl δίνει στην ΚΒΔ την όψη xxx. Αντίστοιχα, οι πίνακες distr δεν αντιγράφονται αλλά η λογική ένωση των πινάκων γγγ αυτού του τύπου σε κάθε κόμβο δίνει στην ΚΒΔ μια όψη γγγ. Οι πίνακες lc1 είναι στην ΚΒΔ απευθείας συνδεδεμένοι με τους αντίστοιχους πίνακες στον κόμβο συντονιστή. Τέλος, οι πίνακες lc11 είναι στην ΚΒΔ απευθείας συνδεδεμένοι με αντίστοιχους κενούς πίνακες με ίδιο αρχικό όνομα και επίθεμα _1 στον κόμβο - συντονιστή.

7.2 Πίνακες αναφοράς

Οι πίνακες αναφοράς ανήκουν στις πληροφορίες εφαρμογών και είναι όλοι τύπου lc1 δηλαδή στην ΚΒΔ είναι απευθείας συνδεδεμένοι με τον αντίστοιχο πίνακα του κόμβου - συντονιστή. Έχουν επίσης μια κοινή μορφή: αν το όνομα του πρωτεύοντος κλειδιού είναι px.xxx το όνομα του πίνακα είναι xxxs ("πληθυντικός" στην Αγγλική γλώσσα). Το πρωτεύον κλειδί είναι το πεδίο από το οποίο παίρνουν τιμή τα εξωτερικά κλειδιά των άλλων πινάκων. Είναι συνήθως τύπου char (8). Ο αλφαριθμητικός τύπος - συμβολοσειρά του πρωτεύοντος

¹² Για κάθε πίνακα ή για κάθε ομάδα πινάκων με κοινά χαρακτηριστικά περιλαμβάνονται κάποια σχόλια και επίσης ο τύπος του πίνακα (rpl, rmt, distr, lc1, lc11). Ο τύπος έχει να κάνει με τον τρόπο που αντιμετωπίζεται το αντικείμενο από την ΚΒΔ, όπως επεξηγήθηκε παραπάνω.

Επίσης καταγράφονται τα πεδία του κάθε πίνακα, η σημασία τους, ο τύπος τους πιθανά σε συντομογραφία (integer1 ή i1, integer2 ή i2, integer4 ή i4, float4 ή f4, float8 ή f8, char(...) ή c(...), varchar(...) ή vc(...), date, hsiid, hdate, coord, ordpair κτλ), το αν είναι nullable (N), το αν είναι πρωτεύοντα (ΠΚ) ή δευτερεύοντα (ΔΚ) κλειδιά { Γ } και η αντίστοιχη ύπαρξη δεικτών και ο τύπος τους (HEAP, BTREE, HASH, ISAM). Αν το κλειδί είναι σύνθετο περιλαμβάνεται και ένας αριθμός που δείχνει τον αύξοντα αριθμό του πεδίου στο κλειδί. Δηλαδή τελικά οι χαρακτηρισμοί των κλειδίων μπορούν να έχουν τη μορφή ΠΚ (HASH, 1), ΠΚ (BTREE, 2), ΔΚ (ISAM), ΠΚ (HEAP) κτλ. Τα κλειδιά, εκτός και αν επισημαίνεται διαφορετικά με το χαρακτηρισμό NU, είναι μοναδικά. Επίσης, αν το πεδίο είναι εξωτερικό κλειδί, ο πίνακας και το πεδίο του από τον οποίο παίρνουν τιμές με τη μορφή EK (table.column).

κλειδιού είναι πολύ πιο ευανάγνωστος και κατανοητός από κάποιο πχ. αριθμητικό κλειδί ενώ και το μέγεθος των 8 χαρακτήρων επαρκεί για να χαρακτηρίσει - ταυτοποιήσει πλήρως το αντικείμενο τόσο στη ΒΔ όσο και στον προγραμματιστή και στον χρήστη. Επίσης υπάρχει και ένα πεδίο τύπου char (40) που περιέχει το πλήρες όνομα του κλειδιού και ονομάζεται full_xxx, ένα integer1 πεδίο num που περιέχει κάποιον αύξοντα αριθμό μοναδικό για αυτή την εγγραφή και ένα πεδίο char (20) το remarks που συνήδως είναι null αλλά μπορεί να περιέχει επιπλέον πληροφορίες για το κλειδί. Το κλειδί είναι πρωτεύον και ο δείκτης έχει τύπο HASH, είναι δηλαδή xxx: ΠΡΩΤ (HASH). Τα περισσότερα κλειδιά είναι τύπου HASH διότι οι πίνακες αυτοί είναι σχετικά ή εξ' ολοκλήρου στατικοί, δηλαδή δεν αλλάζουν, και οι αναζητήσεις γίνονται σχεδόν πάντα με τελεστές ισότητας (πχ. where service = 'EMY'). Αν ο πίνακας ακολουθεί την κοινή μορφή των πινάκων αναφοράς δεν αναγράφονται όλες οι πληροφορίες για τα πεδία του. Στο Παράρτημα Β περιέχονται όλες οι εγγραφές των πινάκων αναφοράς.

Οι πίνακες αναφοράς είναι:

- **acct_ts [1c1]**
Ο πίνακας αυτός περιέχει τις δυνατές τιμές που μπορεί να περάσουν ως παράμετροι στη ΔΒΔ χρέωσης _acct(). Περιέχει δηλαδή (στο πεδίο acct_t) τους δυνατούς χαρακτηρισμούς και τις κατηγορίες των δεδομένων και των τρόπων παρουσίασης και εκτύπωσης.
- **applobj_ts [1c1]**
Οι διαφορετικοί τύποι αντικειμένων στις εφαρμογές Windows/4GL. Πχ. Button, TableField κτλ. Τα πεδία είναι:
applobj_t [c(16), ΠΚ (HASH)]
Ο τύπος αντικειμένου της εφαρμογής.
remarks [c(20), N].
- **aquifers [1c1]**
Περιέχονται οι υδροφορείς (υδρογεωλογικές λεκάνες) της Ελλάδας.
- **categories [1c1]**
Περιέχονται οι δυνατές κατηγορίες (πχ. επιφανειακής υδρολογίας) και υποκατηγορίες (πχ. Σταθμημετρικός) των σταθμών του ΥΔΡΟΣΚΟΠΙΟΥ. Τα επιμέρους πεδία είναι:
category [c(8), ΠΚ (HASH, 1)]
Είναι η κύρια κατηγορία του σταθμού.
full_category [c(40)]
Το πλήρες όνομα της κατηγορίας του σταθμού.
subcategory [c(8), N, ΠΚ (HASH, 2)]
Η υποκατηγορία του σταθμού.

full_subcategory [c(40), N]

Η πλήρης υποκατηγορία του σταθμού.

num [i1]

Αύξων αριθμός.

remarks [c(20), N]

Παρατηρήσεις.

- **codes [lcl]**

Σε μερικές περιπτώσεις, πχ. στην αποθήκευση των χρονοσειρών καιρού, χρησιμοποιούνται κώδικες (μικροί ακέραιοι με συγκεκριμένη τιμή και σημασία). Οι κώδικες που χρησιμοποιούνται περιέχονται στον πίνακα codes. Τα πεδία του είναι:

code [i1, PK(HASH, 1)]

Η τιμή του κώδικα.

code_t [c(8), PK(HASH, 2), EK(code_ts.code_t)]

Το είδος του κώδικα (πχ. 'ΚΑΙΡΟΣ').

wmocode [c(40), N]

Ο πίνακας κωδίκων του WMO από τον οποίο προέρχεται αυτός ο κώδικας (αυτούσιος ή τροποποιημένος). Πχ. WMO 4677 για τον καιρό.

descr [c(40)]

Η λεκτική περιγραφή που αντιστοιχεί σε αυτόν τον κώδικα. Πρέπει να παρουσιάζεται από τις εφαρμογές στο χρήστη αντί για την αριθμητική τιμή του κωδικού.

pathname [c(255), N]

Το πλήρες όνομα του αρχείου στο οποίο υπάρχει εικονική περιγραφή του κώδικα (πχ. για σύμβολα καιρού ΚΑΙΡΟΣ-Σ). Αν έχει νόημα μπορεί να παρουσιάζεται από τις εφαρμογές σε κατάλληλο πεδίο τους.

remarks [c(20), N]

- **code_ts [lcl]**

Οι δυνατοί τύποι κωδίκων.

- **constant_ts [lcl]**

Οι δυνατοί τύποι σταθερών. Περιέχει τα αναμενόμενα πεδία (constant_t, full_constant_t κτλ.) και δυο επιπλέον πεδία:

obj_t [c(8), EK(obj_ts.obj_t)]

Ο τύπος του αντικειμένου (ΣΤΑΘΜΟΣ, ΟΡΓΑΝΟ, ΧΡΟΝΟΣΕΙΡΑ) στο οποίο αναφέρεται η σταθερά.

table_name [c(32)]

Το όνομα του πίνακα στον οποίο αποθηκεύονται τα δεδομένα της σταθεράς.